www.projetoderedes.kit.net

Regras para Proteção de Redes IP

Osmar Ribeiro Leão

OSMAR RIBEIRO LEÃO

REGRAS PARA PROTEÇÃO DE REDES IP

Monografia apresentada ao Programa de Graduação em Informática da Universidade Católica do Salvador, como requisito parcial para obtenção do bacharelado em Informática.
Orientador: Mestre Edeyson Andrade Gomes.

SALVADOR
2001

Resumo

O uso de *firewalls* para defesa de redes de computadores é uma prática bastante comum entre as instituições que desejam proteger seus dados. O uso isolado desta forma de defesa não resolve os problemas, pois são vulneráveis a outras formas de ataque. Nenhuma forma de defesa é totalmente segura. O objetivo do trabalho é apresentar soluções que permitam prevenir e detectar, quando não for possível impedir, ataques e invasões em redes de computadores de todas as formas.

Abstract

A common procedure among institutions that intend to protect and safeguards its data is the use of firewalls as a defense system in computer networks. However, the single use of this kind of defense does not guarantee that the network will be protected against all outsiders attack. Computer networks are still vulnerable to different types of attacks and the current defense system used by many organizations is not completely safe. Therefore, the purpose of this study is to present solutions that may prevent and detect attacks that constantly inflict operational damage in computer networks.

Lista de Tabelas

Lista de Figuras

Sumário

1. Introdução

1.1. Contexto da Pesquisa

Todos os veículos de comunicação quando atingem um patamar global ou de relevância dentro da sociedade necessitam de meios para garantir sua integridade e confiabilidade diante do seu público. Redes de computadores integram este cenário.

O amadurecimento das defesas de rede de computadores é contínuo e leva a erros e equívocos nas soluções e tentativas de proteção em algumas delas. Era muito comum o uso de soluções pela obscuridade, onde eram implantados sistemas proprietários que não mostravam como era feita a defesa e, por tal razão, pensavam que os dados estariam seguros. Ainda hoje é muito usada a defesa em máquinas isoladas, ou seja, protegendo todas as máquinas isoladamente da rede, com isto toda a rede estaria protegida. O pensamento de proteção por máquina isolada é cara e inviável em redes com um número considerável de máquinas, não oferecendo proteção aos dados que trafegam entre as máquinas (no meio físico da rede). O uso mais comum atualmente é o de um ponto centralizado, obrigando o tráfego de rede a passar por ele e tornando este ponto seguro. Com esta forma de proteção é possível proteger a rede, com todos os seus componentes, utilizando *Firewalls*[1], que é o ponto central da rede. O uso de tais sistemas de defesa não é falho, apenas incompleto, necessitando de melhorias para que a defesa seja completa.

Defender não é apenas evitar; quando não for possível impedir uma invasão ou ação imprópria, deve ser possível detectar e alertar o fato.

A defesa completa de redes de computadores vem da união entre o impedimento, prevenção e detecção de falhas, sejam elas de qualquer tipo. É preciso que sejam criadas políticas e que decisões sejam tomadas com a maior rapidez possível, mesmo que meramente informativas.

Criadas as regras de segurança, é então iniciado o processo de detecção a falhas e pontos críticos que devem ser vigiados para manter a confiabilidade perante os usuários dos serviços dependentes da rede de computadores.

---

[1] *Firewall* – Parede de Fogo – Sistemas de defesa de redes mais utilizadas no mercado atual. Os dados passam pelo *firewall* e são submetidos a regras de validação para continuarem a trafegar pela rede.

1.2. Falhas de Segurança em Rede

A detecção de falhas na segurança é o ponto mais complicado nos sistemas de defesa de redes de computadores.

O maior de todos os erros atuais é supor que um único sistema de defesa, só e isolado, vai garantir que toda a rede, onde funcionam diversos tipos de serviços e máquinas com configurações e níveis de segurança diferentes, esteja totalmente segura. Mesmo supondo que este único sistema é imune a falhas, ainda assim não continuará resistindo a crescente evolução dos ataques e erros encontrados nos seus próprios protocolos.

É considerada falha de segurança qualquer acesso a sistemas e informações que não é expressamente autorizado, que não sejam feitas de forma clara – dentro das normas de conduta dos protocolos de redes definidos – e que não estejam de acordo com as regras estabelecidas dentro da instituição proprietária das informações. É também estabelecida como falha a interrupção dos serviços por motivos não definidos, causadas por pessoas ou máquinas não autorizadas, de forma comum ou não.

As maiores falhas na defesa de redes de computadores estão dentro da própria rede, onde as máquinas que deveriam ser protegidas contra ataques externos tornam-se máquinas que atacam outras, levando a inversão do ambiente de defesa. O acesso à rede, mesmo que efetuado de forma válida, pode ser uma violação no sistema de segurança quando é efetuado de maneira suspeita e fora do horário estipulado. A segurança física de certos componentes da rede de computadores também é um fator significativo e relevante na análise da segurança, ou seja: o acesso fácil e irrestrito aos locais onde se encontram máquinas vitais da rede de computadores, bem como a permissão do uso local de dispositivos projetados para serem usados em rede e pela rede, também constituem falha na segurança, possibilitando a interrupção dos serviços.

O acesso aos serviços por meio de suas próprias falhas é visto como falta grave e deve ser tomado cuidado especial, pois nestes casos é necessária a interrupção do serviço. Os erros em protocolos de comunicação têm que ser monitorados exaustivamente, pois não é possível a interrupção do uso do protocolo de comunicação numa rede; as falhas devem ser isoladas e resolvidas com a máxima prioridade. É baseado em falhas nos protocolos que são criados os ataques e invasões mais eficientes, os quais são quase imperceptíveis aos dispositivos de segurança que são construídos

baseados nestes protocolos. Para estes ataques, o estudo e a análise do comportamento da rede levam à conclusão que algo está errado, mal intencionado ou é destrutivo.

É importante para a defesa de redes saber, dentro das requisições de serviços e pacotes de dados[2], quais destes são corretos, incorretos, alarmes falsos ou mecanismos de distração. Um meio de levar um sistema de defesa a falhas e, conseqüentemente, ficar vulnerável, é sobrecarregá-lo com falsos ataques rápidos e simples, desviando a atenção para causas menores e enfraquecendo pontos vitais que podem estar sendo atacados.

Por último, é usado o monitoramento do comportamento da rede. Este meio de defesa é questionado por motivos éticos: até onde chegam os limites do gerente de rede dentro do monitoramento de pacotes e arquivos e seu nível de acesso e intervenção nos documentos e negócios dos usuários.

1.3. Objetivos do Trabalho

O objetivo é organizar uma forma eficiente de defesa de redes, desenvolvendo técnicas para auxílio do *firewall*. A defesa se tornará mais eficaz com o uso de um agente auxiliar que vigiará o comportamento da defesa. Este agente é denominado IDS[3].

Serão construídas novas regras no próprio *firewall* para que seu comportamento seja mais inteligente e preventivo (geralmente os *firewalls* apenas barram os acessos pré-configurados como inválidos). Com as novas regras, será possível dificultar tentativas de varreduras de portas de comunicação, envio de requisições falsas, ataques camuflados, dentre outros. Estas regras já são avanços nos sistemas de defesa de redes. Paralelos ao funcionamento do *firewall*, agentes estarão ativos a procura de padrões de ataques e eventuais quebras dessas regras, alertando todo e qualquer comportamento suspeito ao gerente de rede.

[2] Parte integrante de uma informação. Por limitação do meio físico de uma rede, uma informação é segmentada em pacotes, quadros, etc.

[3] IDS – *Intrusion Detection System* – Sistema de Detecção de Intrusos.

A defesa será evolutiva com a contínua inclusão de novos padrões de ataques e de formas mais eficazes de defesa, permitindo a reestruturação gradativa, coordenada e mais eficiente na defesa de redes.

1.4. Organização da Dissertação

No capítulo 2, serão apresentados os protocolos com o estudo da suas estruturas, funcionamento e apresentação de possíveis falhas e dificuldades no controle a ataques. O estudo será de extrema importância para o entendimento das formas de funcionamento de uma rede.

No capítulo 3, serão apresentadas as formas de defesas, como *firewalls*, filtro de pacotes, arquiteturas de defesa, servidores *proxy*[4] e a detecção e prevenção a invasões e ataques coordenados. Este capítulo contém os princípios de defesa utilizados na defesa de redes e demonstram quais são os ataques e comportamentos que os agentes de defesa procuram.

No capítulo 4, será mostrado o estudo de um caso, detalhando a estrutura da rede e apresentando as regras para a sua defesa (Anexo E).

No capítulo 5, serão mostradas as conclusões do trabalho e a tendência evolutiva na defesa de redes de computadores.

---

[4] Do inglês, Procurador. Agente que gerencia e intermedeia a comunicação de um determinado serviço de rede entre duas máquinas em redes distintas.

2. Protocolos

2.1. IP

2.1.1. Conceitos

Internet Protocol, Protocolo da Internet ou Protocolo Internet, é o nome de um dos protocolos de comunicação mais usados no mundo em cerca de 90% do tráfego da rede mundial de computadores – Internet – atualmente.

O IP está no terceiro nível da camada de rede do modelo de camadas OSI[5] (está na segunda camada no modelo de camadas Internet), podendo trafegar sobre vários protocolos de comunicação da segunda camada deste modelo. Está definido em forma de RFC[6] de número 791 e foi criado pelo Departamento de Defesa do Governo dos Estados Unidos para comunicação entre bases militares no conjunto de protocolos de interconexão do projeto DARPA[7].

É um serviço de comunicação sem conexão direta dos dois pontos finais (emissor e receptor), havendo várias rotas para a ligação. Por causa deste fator, é possível que haja gargalos de comunicação, gerando engarrafamento e perda de pacotes pelos motivos do próprio tráfego ou erros no envio. Outro fator que pode ocorrer por não ser orientado a conexão é a chegada de pacotes fora de ordem, quando o datagrama[8] IP é fragmentado. A falta de controle de fluxo também constitui uma de suas deficiências.

Por estar numa camada acima da camada de enlace (modelo OSI), o IP oculta as redes (que podem ter diferentes protocolos e tamanhos de quadro[9]) pelas quais ele passa, facilitando sua instalação e tornando-o mais robusto.

---

[5] OSI – *Open Systems Interconnection* – camadas que definem padrões para comunicação em uma rede de computadores.

[6] RFC – *Request for Comments* – documentos que definem um determinado protocolo de internet ou experimento relacionado. Os RFCs são mantidos pelo Instituto de Engenharia Elétrica e Eletrônica, IEEE.

[7] *Defense Advanced Research Projects Agency* – Agência de Projetos de Pesquisas Avançados sobre Defesa. Órgão do Exército dos Estados Unidos da América.

[8] Pacote de dados IP. IP *datagram*.

[9] Unidade de medida utilizada na segunda camada do modelo OSI: quadros, *frames* na língua inglesa.

Endereçamento

O IP usa endereços para identificar máquinas e redes. Estes endereços são expressos de forma binária e, para o melhor entendimento pelo homem, existem as respectivas representações por quatro números inteiros decimais separados por pontos e apelidos equivalentes aos endereços binários.

No IP, cada máquina tem um endereço único na rede que é um número inteiro de 32 bits denominado endereço IP. Para todos os tipos de comunicação com esta máquina é necessária a utilização deste endereço.

Conceitualmente, o endereço IP é constituído por um par (*netid*, *hostid*), onde *netid* é o identificador de rede do endereço, enquanto o *hostid* é o identificador da máquina nesta rede.

Existem três classes principais de endereços de redes IP: Classes A, B e C. A classe do tipo A é usada para uma rede que tenha um número grande de máquinas, 2^{16} (65.536 máquinas), reservando sete bits para o identificador de rede e 24 bits para identificador de máquinas. A classe tipo B é usada para uma rede que tenha um número médio de máquinas, entre 2^{8} (256 máquinas) e 2^{16}, reservando 14 bits para o identificador de rede e 16 bits para o identificador de máquinas. A classe tipo C é usada para redes com não mais de 2^{8} máquinas, reservando 21 bits para o identificador de rede e somente oito bits para o identificador de máquinas. Ainda existem as classes D e E que são usadas para propósitos específicos.

Para cada rede é necessário um endereço IP válido dentro daquela rede. Quando a máquina tem mais de uma conexão física é necessário um endereço IP para cada conexão de rede; assim, uma máquina conectada a 10 redes tem 10 endereços diferentes de IP.

Existem, ainda, os endereços de redes – que servem para identificar a rede de endereços IP – e os endereços de difusão – que servem para propagar uma mensagem para todas as máquinas nesta rede. O endereço da rede é aquele no qual o *hostid* é zero. Ao contrário disto, no endereço de difusão todos os bits do *hostid* são iguais a um. Na Tabela 1 é mostrado um exemplo de endereços IP numa rede de classe tipo C.

Endereço	Tipo	Notação Binária do *hostid*
192.168.1.0	Endereço de rede	00000000
192.168.1.1	Endereço comum	00000001
192.168.1.2	Endereço comum	00000010
192.168.1.254	Endereço comum	11111110
192.168.1.255	Endereço de difusão	11111111

Tabela 1 – Endereços de Rede e Difusão.

É possível ainda delimitar uma rede diminuindo-a em sub-redes. As sub-redes são redes sem classe e podem ser criadas baseadas na informação do endereço de difusão diferentes. Estes endereços são delimitados pelas máscaras de difusão de rede ou máscaras de rede.

Outra forma de enviar mensagens para muitas máquinas numa rede é utilizar uma particularidade do IP chamada de *multicasting*. O IP *multicasting* utiliza endereços classe D para criar grupos e, assim, fazer a difusão de mensagens entre essas máquinas. Os endereços de classe D têm os quatro primeiros bits em 1110 e os 28 bits restantes identificam os grupos *multicast*. Com esta configuração de endereços é possível haver endereços IP para multcasting entre 224.0.0.0 até 239.255.255.255. O endereço 244.0.0.1 é reservado para um grupo com todas as máquinas e a rede internet de computadores não aceita endereços de *multicast* 224.0.0.0.

As máquinas que desejam participar de grupos *multicast* necessitam enviar, em um protocolo específico, mensagem para o endereço de *multicast* desejado. Este protocolo chama-se IGMP (*Internet Group Management Protocol*), do inglês, Protocolo de Gerenciamento de Grupos Internet. Com este protocolo é possível ingressar em grupos, sair dos grupos e enviar mensagens especiais entre os grupos.

O IP m*ulticast* é usado em larga escala para a entrega de conteúdo multimídia, como vídeo e áudio, e também, está crescendo o seu uso para voz sobre IP.

Pacote IP

O datagrama IP é como está divido um pacote IP e suas especificações são mostrados na Figura 1.

Version (4)	*Header Length (4)*
Type of Service (8)	
Total Length (16)	
Identifier (16)	
Flags (3)	*Fragment Offset (13)*
Time to Live (8)	
Protocol (8)	
Header Checksum (16)	
Source Address (32)	
Destination Address (32)	
Options and Padding (Variable)	
Data (Variable)	

Figura 1 – Datagrama IP.
(n) = Número de bits no campo.

O campo *Version* – Versão – identifica qual a versão do protocolo IP utilizada[10]. Este campo faz parte do cabeçalho.

O campo *Header Lenght* – Tamanho do Cabeçalho – é medido em palavras de 32 bits. Tipicamente um cabeçalho sem opções de Qualidade de Serviço (apresentado logo abaixo) tem 20 octetos[11]; portanto, o valor geralmente é de 5: $2^5 = 32$. Este campo faz parte do cabeçalho.

O campo *Type of Service* – Tipo de Serviço – é usado para identificar as funções de Qualidade de Serviço. O tipo de serviço é formado por cinco entradas totalizando o campo de 8 bits (Figura 2). Os três primeiros bits contêm o valor de precedência (por exemplo: 000 representa precedência de rotina, 111 é usado em algumas

[10] Para todas as referências usadas neste, a versão do protocolo IP sempre será a número 4.

[11] Um octeto é uma palavra de 8 bits.

implementações para indicar controle de datagramas de rede). Os bits de precedência podem ser usados para controle de fluxo e mecanismos de controle de congestionamento, fazendo com que máquinas decidam sobre a ordem de descarte de pacotes. O quarto bit é usado como bit de atraso (D bit); quando está em 1, o Tipo de Serviço requisita um pequeno atraso na rede. O quinto bit é usado como o bit de passagem (T bit); quando está em 1, requisita rápida passagem pela rede. O sexto bit é usado como o bit de confiança (R bit), que permite que o usuário solicite grande confiança do datagrama. Os bits sete e oito não são utilizados. Este campo pode ser usado na implementação de melhorias de rotas e caminhos na conexão IP. Este campo faz parte do cabeçalho.

Figura 2 – Os cinco sub campos que compõem o campo Tipo de Serviço.

O campo *Total Length* – Tamanho Total – especifica o tamanho total do datagrama IP. Para o cálculo do tamanho do cabeçalho, basta subtrair o tamanho do cabeçalho do tamanho total para obter o tamanho da parte de dados. O tamanho máximo possível para um datagrama IP é de 65.535 octetos (2^{16}). Este campo faz parte do cabeçalho.

O campo *Flags* – Bandeiras – é usado no controle de fragmentação e remontagem dos datagramas IP. É medido em octetos. Os campos são: identificador (*identifier*), bandeiras (*flags*) e compensação de fragmentos (*fragment offset*). O identificador é usado para identificar unicamente todos os fragmentos de um datagrama original. É utilizado com o endereço fonte (*source address*) para identificar cada fragmento na máquina de destino. O campo bandeira (*flag*) é utilizado para indicar se o datagrama pode ou não ser fragmentado. Podendo ser fragmentado, o último bit do campo determina se o fragmento é o último do datagrama. O campo de compensação de fragmentos determina a posição relativa do fragmento em relação ao datagrama original; é iniciado em zero e acrescentado conforme o número de fragmentos gerados. Este campo faz parte do cabeçalho.

O campo *Time to Live* – Tempo de Vida – é usado para calcular o tempo de vida que o datagrama IP está trafegando. Seu valor é contado em ordem inversa e quando é

atingido o tempo final (zero), e o pacote ainda não atingiu o destino, o próximo roteador da rota descarta o datagrama IP. Este processo é feito a cada passagem por um roteador, diminuindo o valor em um. O tempo de Vida é utilizado para calcular o número de pulos (*hops*) do datagrama IP na sua rota entre a máquina emissora e a receptora. Além disso, serve para a prevenção de voltas intermináveis na rota do datagrama (*loop*).

O campo de *Protocol* – Protocolo – é usado para indicar o próximo protocolo na camada de rede OSI. Os grupos responsáveis pela padronização da Internet criaram os números padrões de cada protocolo (Tabela 2). Este campo faz parte do cabeçalho.

Decimal	Sigla	Protocolo
0	-	*Reserved*
1	ICMP	*Internet Control Message Protocol*
2	IGMP	*Internet Group Management Protocol*
3	GGP	*Gateway-to-Gateway Protocol*
4	-	*Unassigned*
5	ST	*Stream*
6	TCP	*Transmission Control Protocol*
7	UCL	*UCL*
8	EGP	*Exterior Gateway Protocol*
9	IGP	*Interior Gateway Protocol*
10	BBN-MON	*BBN-RCC Monitoring*
11	NVP-II	*Network Voice Protocol*
12	PUP	*PUP*
13	ARGUS	*ARGUS*
14	EMCON	*EMCON*
15	XNET	*Cross Net Debugger*
16	CHAOS	*Chaos*
17	UDP	*User Datagram Protocol*
18	MUX	*Multiplexing*
19	DCN-MEAS	*DCN Measurment Subsystems*
20	HMP	*Host Monitoring Protocol*
21	PRM	*Packet Radio Monitoring*
22	XNS-IDP	*Xerox NS IDP*
23	TRUNK-1	*Trunk-1*
24	TRUNK-2	*Trunk-2*
25	LEAF-1	*Leaf-1*
26	LEAF-2	*Leaf-2*
27	RDP	*Reliable Data Protocol*
28	IRTP	*Internet Reliable TP*
29	ISSO-TP4	*ISO Transport Class 4*
30	NETBLT	*Bulk Data Transfer*
31	MFE-NSP	*MFE Network Services*
32	MERIT-INP	*MERIT Internodal Protocol*
33	SEP	*Sequencial Exchange*
34-60	-	*Unassigned*
61	-	*Any Host Internal Protocol*
62	CFTP	*CFTP*
63	-	*Any Local Network*
64	SAT-EXPAK	*SATNET and Backroom EXPAK*
65	MIT-SUBN	*MIT Subnet Support*
66	RVD	*MIT Remote Virtual Disk*
67	IPPC	*Internet Plur. Packet Core*
68	-	*Any Distributed File System*
69	SAT-MON	*SATNET Monitoring*
70	-	*Unassigned*
71	IPCV	*Packet Core Utility*
72-75	-	*Unassigned*
76	BRSAT-MON	*Backroom SATNET Monitoring*
77	-	*Unassigned*
78	WB-MON	*Wideband Monitoring*
79	WB-EXPAK	*Wideband EXPAK*
80-254	-	*Unassigned*
255	-	*Reserved*

Tabela 2 – Número de Protocolos Internet

O campo *Header Checksum* – Verificação de Soma de Cabeçalho – é usado para detectar qualquer distorção no cabeçalho IP. Existe um algoritmo simples para detecção de erro que é efetuado pelo roteador. Caso haja um erro, é enviada de volta ao emissor uma notificação de erro, utilizando o protocolo ICMP[12] (Capítulo 2, Seção 2). Este campo faz parte do cabeçalho.

Os campos *Source Address* – Endereço Fonte – e *Destination Address* – Endereço de Destino – guardam os endereços IP do emissor e do receptor. Eles permanecem inalterados até o fim do envio. Estes campos fazem parte do cabeçalho.

O campo *Options* – Opções – é usado para tipos de opções adicionais e é utilizado de formas variadas pelos fabricantes de *software* e *hardware*. Este campo faz parte do cabeçalho.

O campo *Padding* – Preenchimento – é usado para ajustar o tamanho do datagrama no limite de 32 bits, preenchendo o restante do datagrama com zeros para que a soma chegue a 32 bits. Este campo faz parte do cabeçalho.

O campo *Data* – Dados – é onde estão contidos os dados do usuário, as informações que estão sendo passadas. A combinação entre dados e cabeçalho não pode ultrapassar 65.535 octetos.

2.1.2. Fragmentação

O datagrama IP passa sobre tipos de protocolos diferentes do segundo nível da camada OSI, que, na maioria dos casos, têm tamanho de quadros diferentes. Para que o datagrama IP possa trafegar sem problemas sobre estes protocolos é necessária uma forma de fragmentação e remontagem do datagrama IP.

Todos os protocolos de redes têm um tamanho máximo de PDU[13], chamado de Unidade Máxima de Transmissão (*maximum transmission unit*: MTU). Quando um roteador recebe um datagrama que tem tamanho maior que um MTU, ele fragmenta o datagrama para ajustá-lo ao tamanho do MTU. Todos os fragmentos têm cabeçalho e parte de dados contendo no cabeçalho informações pertinentes à fragmentação. Cada fragmento pode tomar rotas diferentes para chegar ao destino, agindo como datagramas

---

[12] *Internet Control Message Protocol* – Protocolo de Mensagem de Controle – utilizado para envio de códigos de mensagem padrão.

[13] *Protocol Data Unit* – Unidade de dados do Protocolo – é a unidade de medida de um protocolo. Exemplo: 3 quadros, 3 PDUs.

comuns e causando a chegada de alguns fragmentos primeiro do que outros e fora de ordem.

Como cada fragmento do pacote pode tomar caminhos diferentes, utilizando o tamanho máximo de MTU estipulado pela última parte do caminho (pelo roteador), é possível que o fragmento sofra mais uma fragmentação, tornando-o ainda menor (Figura 3). Como o IP contém campos de controle de fragmentação, posição relativa e tamanho do pacote, é possível ser feito o cálculo de remontagem dos fragmentos, mesmo quando eles são re-fragmentados e recebidos fora de ordem.
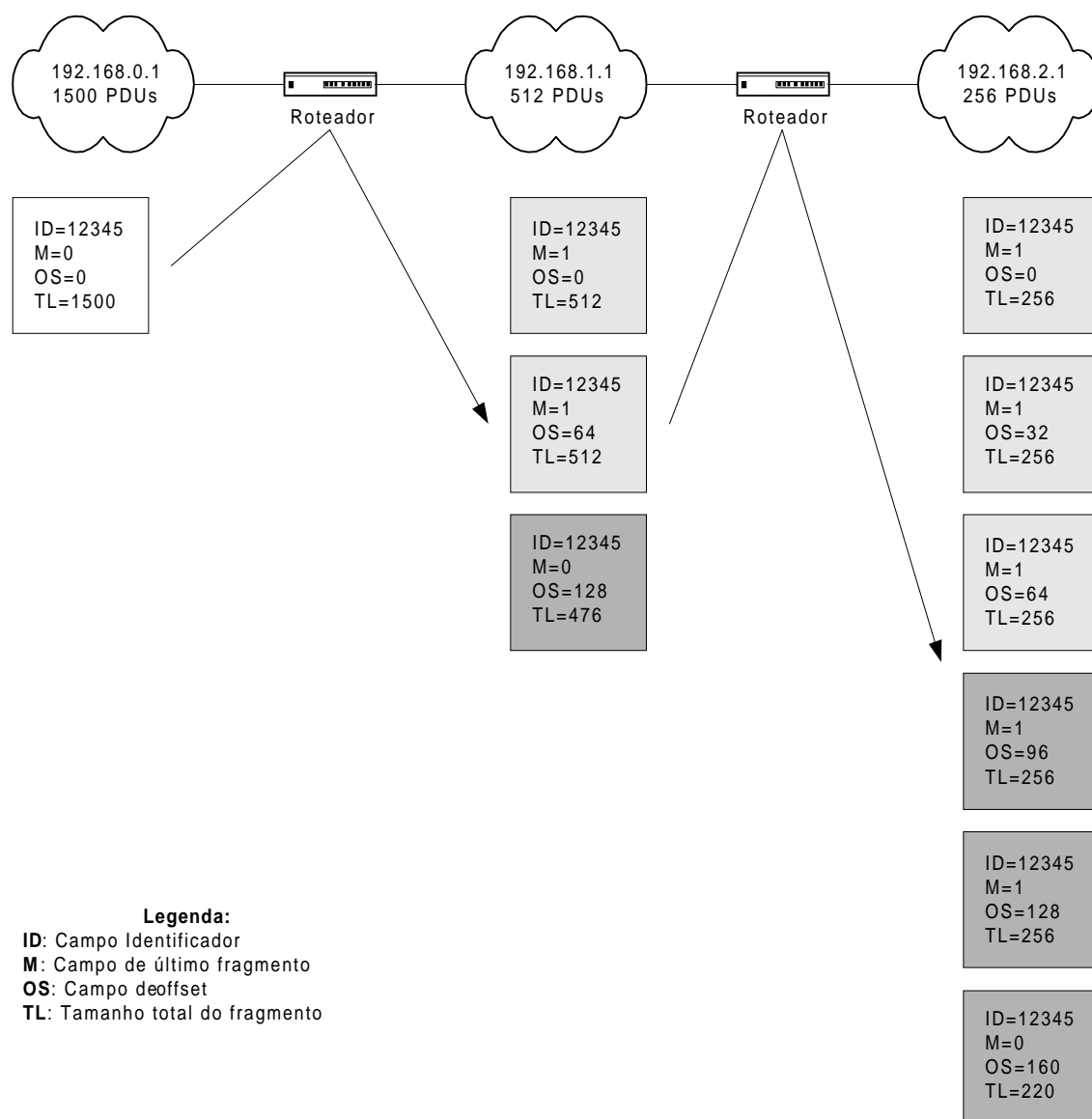
Figura 3 – Operações para fragmentação em Roteadores.

A máquina receptora tem que calcular o tamanho do pacote para remontá-lo. O receptor só saberá qual o tamanho do pacote quando chegar ao *buffer* de recepção o último pacote da fragmentação, onde a *flag* de último fragmento está indicada, que não é obrigatoriamente o último dos fragmentos do pacote. Com o último fragmento, é possível calcular o tamanho máximo do pacote sem todos os fragmentos simplesmente utilizando o bit de compensação (*offset*), multiplicando por oito e somando com os valores indicados no campo de tamanho total de fragmento.

Na Figura 3, o pacote foi dividido em seis fragmentos. Considerando que o quarto fragmento que chegou ao receptor é o último fragmento do pacote (ainda faltam chegar dois fragmentos), já é possível calcular o tamanho total. O tamanho máximo do último PDU foi de 256 e o último fragmento indica o tamanho de 220 octetos. O receptor, então, calcula o tamanho máximo da seguinte maneira: 160 que é o valor de compensação do último fragmento (seis fragmentos: 0 – 32 – 64 – 96 – 128 - 160) multiplicado por 8 (um octeto) mais os 220 octetos finais que vieram no fragmento, totalizam 1500 octetos (tamanho máximo do datagrama IP).

O campo de Tamanho de Campo Total referencia o tamanho total do fragmento, não do pacote, por isso é tão necessário o bit do último fragmento.

Se, por qualquer razão, algum fragmento é descartado no caminho, o receptor descarta os fragmentos do pacote que chegaram. Existe, ainda, o tempo máximo estipulado pelo gerente da rede para que os pacotes cheguem ao receptor (Tempo de Vida - TTL). Caso o tempo se esgote e algum dos fragmentos ainda não tenha chegado, o receptor descarta todos os fragmentos anteriores do pacote. Este tempo pode ser alterado.

2.1.3. Roteamento

O roteamento serve para indicar, quando existe um ou mais dispositivos de rede numa máquina, qual caminho o pacote deve seguir ou por onde ele deve ser despachado. Existem as tabelas de endereços, que indicam ao Sistema Operacional os roteadores e endereços IPs que estão ao alcance da máquina, e as tabelas de roteamento, que indicam como se chegar ao endereço IP desejado (caminho) e por onde deve ser despachado o pacote (dispositivo).

A tabela de endereço é representada da seguinte forma (Figura 4):

```
eth0      Encapsulamento do Link: Ethernet  Endereço de HW 00:00:21:C2:10:B4
          inet end.: 10.18.10.1  Bcast:10.18.10.7  Masc:255.255.255.248
          UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1

hdlc0     Encapsulamento do Link: Protocolo Ponto-a-Ponto
          inet end.: 10.22.11.250  P-a-P:10.22.11.249  Masc:255.255.255.252
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Métrica:1

ppp0      Encapsulamento do Link: Protocolo Ponto-a-Ponto
          inet end.: 10.23.151.1  P-a-P:10.23.151.2  Masc:255.255.255.252
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Métrica:1

lo        Encapsulamento do Link: Loopback Local
          inet end.: 127.0.0.1  Masc:255.0.0.0
          UP LOOPBACKRUNNING  MTU:3856  Métrica:1
```

Figura 4 – Tabela de endereçamento do cerne do Sistema Operacional Linux.

O Endereço é o endereço IP para este dispositivo – *inet end.*

O Índice de Interface é o número do dispositivo pertencente a esta linha na tabela – *eth0, hdlc0, ppp0, lo.*

A Máscara de Rede é a máscara de rede associada ao endereço IP da linha – *Masc.*

O Endereço de Difusão é o endereço que contém o bit menos significativo no endereço IP de difusão. Está associado ao endereço IP da linha – *Bcast.*

O Tamanho máximo é o valor máximo que o datagrama IP pode ter – *MTU.*

Existem ainda outras informações referentes a outras camadas, como o *Endereço de Hardware* e informações sobre métrica e modo do funcionamento da rede (*UP BROACASTRUNNING MULTICAST*).

A tabela de roteamento é representada da seguinte forma (Figura 5):

```
Tabela de Roteamento IP do Kernel
Destino         Roteador        MáscaraGen.     Opções Métrica Ref    Uso Iface
10.18.10.0      0.0.0.0         255.255.255.248 U      0       0        0 eth0
192.168.0.0     0.0.0.0         255.255.255.0   U      0       0        0 hdlc0
200.136.2.50    0.0.0.0         255.255.0.0     U      0       0        0 ppp0
10.19.2.0       10.18.10.1      255.255.255.248 G      1       0        0 eth0
192.168.1.0     192.168.0.1     255.255.255.0   G      1       0        0 hdlc0
127.0.0.0       0.0.0.0         255.0.0.0       U      0       0        0 lo
0.0.0.0         10.18.10.1      0.0.0.0         G      0       0        0 eth0
```

Figura 5 – Tabela de roteamento do cerne do Sistema Operacional Linux.

O Destino contém o endereço IP da máquina ou rede de destino. Caso o endereço seja o 0.0.0.0, esta é considerada a rota padrão.

O Índice de Interface é o número do dispositivo que será despachado o pacote IP referente à rota – *Iface*.

A Métrica é o número de pulos (*hops*) para que seja alcançado o endereço de destino. Hoje quase não é mais usado.

O Endereço do Roteador é o endereço do próximo ponto que o pacote irá chegar; é o próximo pulo do pacote IP. Lá é decidido o próximo e assim sucessivamente até o endereço final.

A Máscara é o endereço da máscara de rede onde o dispositivo está ligado.

O Campo de Opções é o campo que define se o datagrama IP é entregue diretamente à rede ou se enviado ao roteador (U ou G).

Existem, ainda, os campos de Referência (*Ref*) e *Uso* que são meramente informativos ao uso e referência pelo Sistema Operacional do dispositivo.

2.2. ICMP

2.2.1. Conceitos

Para possibilitar que o protocolo IP reportasse os erros no envio, perda de pacotes, necessidades de repetição de envio, melhorias no roteamento de pacotes, dentre outras situações, foi criado o ICMP.

ICMP, do inglês *Internet Control Message Protocol* – Protocolo Internet de Controle de Mensagens – é utilizado para as notificações no nível IP da camada de comunicação. Inicialmente criado para ser usado somente por roteadores na correção de erros de envio, o ICMP, hoje, é também utilizado por máquinas numa rede.

A notificação de uma eventual anormalidade ou um simples alerta não faz do ICMP um protocolo de correção de erros. Apesar de suas notificações serem usadas para tal finalidade, ele é utilizado objetivando informar, dentro das propriedades do IP, situações que necessitam de alguma alteração ou re-configuração. Seu escopo é limitado e não é possível sua utilização para correção do roteamento por roteadores intermediários no percurso do pacote, uma vez que sua mensagem sempre é destinada ao emissor.

Uma mensagem ICMP é um datagrama IP normal, ou seja, está encapsulado dentro de um pacote IP contendo cabeçalho e parte de dados definidos. O pacote IP contendo informações ICMP está mostrado na Figura 6.

Figura 6 – Formato da mensagem ICMP – (n) = Número de bits no campo.

No Cabeçalho IP, como mostrado na tabela de protocolos anterior (Capítulo 2, Seção 1), o campo de protocolo é ajustado em 1.

O campo Tipo é usado para definir o tipo de mensagem ICMP. Este campo faz parte do cabeçalho ICMP.

O campo Código é usado para definir o código de erro ou código de informação referente ao tipo de mensagem. Este campo faz parte do cabeçalho ICMP.

O campo Verificação de Soma é usado para o cálculo de verificação do cabeçalho, que permite verificar erros durante o tráfego. É utilizado o mesmo algoritmo de verificação do IP, só que aplicado ao cabeçalho ICMP. Este campo faz parte do cabeçalho ICMP.

O campo Parâmetros é utilizado por alguns tipos de mensagens ICMP como forma de uso de informações extras.

O campo de Informação é utilizado em notificações de erros com parte do cabeçalho e 64 bits de dados do pacote IP que ocasionou na notificação.

Para que não fosse gerada uma infinidade de mensagens de erros, congestionando ou piorando o congestionamento da rede, é definido na implementação do protocolo ICMP que não são utilizadas mensagens de erros para notificar erros no próprio protocolo.

2.2.2. Tipos de Requisições ICMP

O ICMP apresenta os seguintes tipos de mensagens[14] de erro:

Campo Tipo	Tipo de Mensagem ICMP
0	Resposta ao Eco (*Echo Reply*)
3	Destino não Acessível
4	Dissipação da Origem
5	Re-direcionamento (mudar a rota)
8	Solicitação de Eco (*Echo Request*)
11	Tempo Excedido para um Datagrama
12	Problema de Parâmetro num Datagrama
13	Solicitação de Indicação de Hora
14	Resposta de Indicação de Hora
15	Solicitação de Informação (obsoleto)
16	Resposta de Informação (obsoleto)
17	Solicitação de Máscara de Endereço
18	Resposta de Máscara de Endereço

Tabela 3 – Códigos de Tipos ICMP.

A Solicitação e a Resposta ao Eco – similar ao efeito de eco do som, é a resposta a um sinal na rede, ou seja, simplesmente o retorno do sinal pelo alvo (máquina) – são utilizadas para verificação de vida e funcionamento do receptor. É enviada uma solicitação de eco ao receptor, que por sua vez envia uma resposta ao eco acrescido das informações enviadas dentro da parte de dados do pacote ICMP. Alguns Sistemas Operacionais geram um número em seqüência, contam e formulam estatísticas de perda de pacotes utilizando o ICMP tipo zero e oito, respectivamente, Solicitação e Resposta ao Eco.

O Destino não Acessível é utilizado para informar à máquina que enviou o pacote ou a um roteador que o destino não está acessível por problemas de máquina, rede, protocolos, portas de comunicação não acessíveis, indisponíveis ou desconhecidas.

A Dissipação da Origem (*Source Quench*) é utilizada como forma de notificação de descarte de pacotes por congestionamento por parte de um roteador que esteja muito carregado. As mensagens de *source quench* podem ter duas origens: (1) meios físicos mais lentos do que as máquinas da rede gerando incapacidade do roteador; (2) acúmulo de mensagens devido à falta de poder de processamento pelo roteador para aquele nível de mensagens.

[14] Ainda existem mais detalhes sobre os tipos de mensagens ICMP. O anexo A contém uma lista mais detalhada sobre as mensagens ICMP e os seus tipos.

O Re-direcionamento de rotas é utilizado entre roteadores para notificar uma mudança nas rotas para máquinas específicas. A mudança nas redes também gera mensagens entre roteadores para mudança de rota; porém o uso mais comum deste serviço é para aviso de rotas melhores para determinada máquina vindas do roteador numa rede. É comum o uso deste serviço de modo que sejam otimizadas as rotas para algumas redes e máquinas específicas dentro de uma configuração favorável de ambiente (isto não faria sentido dentro de uma rede simples sem caminhos para outras redes ou outros roteadores).

O Tempo Excedido para um Datagrama é a notificação de um dos roteadores da rota de percurso do pacote IP do emissor ao receptor que, por algum motivo, tenha o Campo de Tempo de Vida expirado. Isso é comum quando há uma má configuração de rotas para uma determinada rede ou máquina, que acarreta em um caminho muito extenso. Serve também, para alertar sobre o problema de *loops* do pacote entre roteadores.

Ainda existem outros tipos de serviços que são para sincronia de hora entre máquinas, solicitação do endereço de máscara de rede para uma resposta, estimativas de tempo de trânsito de pacotes, informação de tamanho ou parâmetro incorreto no pacote IP, dentre outros, que estão detalhados no Anexo A.

2.3. TCP

2.3.1. Conceitos

Na conexão de rede entre duas máquinas, em algumas situações, faz-se necessário o uso de controle de tráfego, correção de erros e os outros tipos de situações. Na primeira seção deste capítulo, foi citado que o protocolo IP não fazia o controle destas situações. De fato, o controle de erros, tratamento de fluxo, correção de erros e outras funções relacionadas são feitos utilizando o protocolo que está intimamente ligado ao IP. O protocolo referido é o TCP (*Transmission Control Protocol*), do inglês, Protocolo de Controle de Transmissão.

O TCP, juntamente com o UDP (Capítulo 2, Seção 4), está na quarta camada no modelo de camadas OSI, que é denominada camada de transporte. Este é o principal protocolo para as tarefas de controle de fluxo do IP e retransmissão nos casos de erros e perdas nos dados.

Como é um protocolo orientado a conexão, ele garante a troca de informações entre os dois pontos da comunicação (emissor e receptor), utilizando-se desta particularidade para realizar as suas principais funções, apesar de estar sobre o IP.

O uso dos dois protocolos juntos – IP e TCP – forma a combinação mais conhecida na rede internet mundial: TCP/IP. Por serem os dois principais protocolos de comunicação na internet, a maioria dos serviços foram escritos e funcionam utilizando as implementações e funcionalidades do TCP/IP.

O protoloco TCP é utilizado em diversos serviços para camadas superiores de comunicações, onde os mais utilizados são: (1) Gerenciamento orientado à conexão, (2) Transferência de dados com segurança, (3) Transferência orientado ao fluxo, (4) Funções de transmissões imediatas, (5) Re-seqüência de envio, (6) Controle de fluxo, (7) Multiplexação, (8) Transmissões *full-duplex*[15], (9) Precedência e segurança e (10) Encerramento de conexões.

O TCP, por ser um protocolo, tem sua própria denominação para o PDU que é conhecida como segmento, resultado do acúmulo de dados recebidos de protocolos superiores. A formação de um segmento vem dos *bytes* provenientes deste acúmulo de informação. Quando é formada uma certa quantidade de informação suficiente é enviado um segmento pela rede.

O controle de fluxo entre as duas máquinas numa conexão TCP é dada da seguinte maneira: para cada segmento enviado para o receptor, é enviado, em anexo, um número de seqüência. Quando os dados e o número de seqüência chegam ao receptor, este envia de volta para o emissor uma informação de conhecimento da chegada (ACK). Se uma informação de confirmação demora a chegar, o emissor manda novamente o segmento e o número de seqüência. Com isto, é possível tratar erros de envio e demora no recebimento. Uma informação sobre o tamanho do segmento (chamado de janela) é trocada entre as duas máquinas que fecharam uma conexão TCP, negociando, assim, qual será o tamanho das informações que trocarão.

Pela particularidade de acúmulo das informações (*bytes*) antes do envio pelo TCP, foi criada uma função para que dados importantes fossem enviados ao receptor imediatamente quando chegassem à camada TCP. Esta função é denominada *push*, do inglês, empurrão.

---

[15] Tipo de transferência de dados que utiliza o mesmo meio físico para transporte de dados nos dois sentidos simultaneamente: do receptor ao emissor e vice-versa.

O TCP é também responsável pela re-seqüência das informações quando estas chegam de forma desordenada ao receptor, assim como por descartar pacotes que, por algum motivo, cheguem duplicados ao receptor.

Sockets

O TCP implementa o conceito de portas para orientar uma conexão entre dois pontos. Porta é uma abstração, dentro do endereço IP, ao serviço ou aplicação de destino ou fonte dos dados. Um exemplo disto é o serviço de e-mail, que utiliza a porta 25, numa determinada máquina, para envio e recepção de mensagens.

O *Socket* é a concatenação entre o endereço IP e uma porta de comunicação. Um par de *Sockets* identifica unicamente cada conexão numa rede. O *Socket* de envio é o endereço IP fonte mais número de porta fonte, enquanto o *Socket* de recebimento corresponde ao endereço IP de destino mais número de porta de destino.

Portas Reservadas e Conexões

Existem diversos tipos de serviços associados (ANEXO B) a portas. As portas abaixo de 255 são chamadas de portas conhecidas e são reservadas para alguns serviços padrão de comunicação TCP/IP.

Para uma máquina conectar-se a outra é necessário que ela reserve uma porta fonte (negociado com o Sistema Operacional) e especifique uma porta de destino, juntamente com o IP da máquina de destino. Foi convencionado que portas abaixo de 1024 são utilizadas somente para recebimento de conexões, com exceções em casos especiais, onde é necessária a utilização das portas baixas[16].

É possível que uma máquina servidora receba várias requisições de conexão na mesma porta, fazendo com que o TCP use sua funcionalidade de multiplexação de conexões para organizar as diversas conexões na mesma porta. O TCP usa o conjunto de endereço IP e porta (*socket*) para identificar diferentes conexões com portas de destino e origem iguais, porém com endereços IP diferentes.

Ainda existem duas formas de abertura de conexão denominada de Passiva e Ativa.

[16] Em casos onde um *firewall* não permite o uso de qualquer porta para conexão.

Na forma Passiva de conexão, o protocolo TCP requisita ao Sistema Operacional uma porta e permanece em estado de espera. Quando uma conexão chega à máquina, o Sistema Operacional repassa a requisição à aplicação que requisitou aquela porta em espera.

Na forma Ativa de conexão, o protocolo TCP requisita ao Sistema Operacional uma porta para recebimento de conexão, porém requer também que seja feita uma conexão para um endereço IP indicado na porta especificada. Na primeira resposta de conexão do endereço IP de destino, o Sistema Operacional repassa a resposta à aplicação que requisitou, na porta que foi requerida.

As operações de envio de mensagens e controle do tamanho de janela e fluxo são descritas nas Figuras 7 e 8.
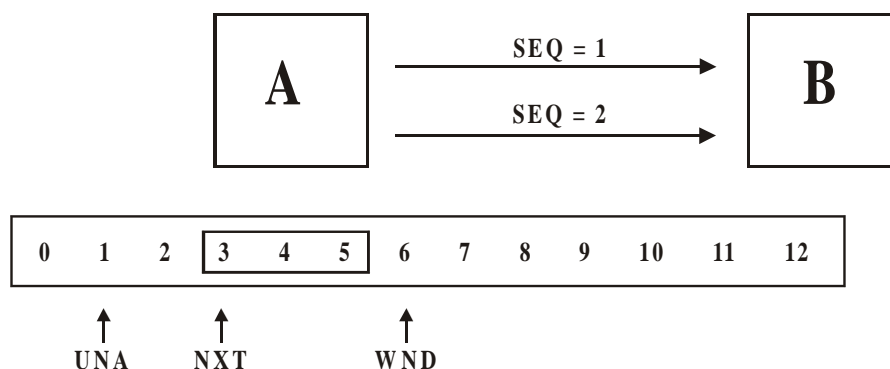
Figura 7 – TCP Envia Variáveis da Janela.

Na Figura 7, o módulo A e B representam duas máquinas numa rede trocando informações utilizando o protocolo TCP.

O módulo A transmite dois segmentos para o módulo B; estes segmentos são rotulados como SEQ = 1 e SEQ = 2. Os ponteiros das variáveis UNA, NXT e WND indicam alguns estados para este envio. Na esquerda do ponteiro UNA, estão os *bytes* que foram enviados e foram confirmados por B (*byte* 0). Os *bytes* entre os ponteiros UNA e NXT (*byte* 1 e 2) são os *bytes* que foram enviados e não confirmados por B. O ponteiro NXT identifica o número da seqüência do próximo octeto a ser enviado. O indicador de limite do tamanho da janela é o indicador do tamanho máximo a ser enviado antes da janela ser fechada. O limite da janela é calculado pela soma entre UNA

e NXT, neste caso é 5 (UNA = 2 e NXT = 3). Como A transmitiu 1 e 2, só poderão ser transmitidos os *bytes* 3, 4 e 5. Esta área está delimitada pelo retângulo menor.
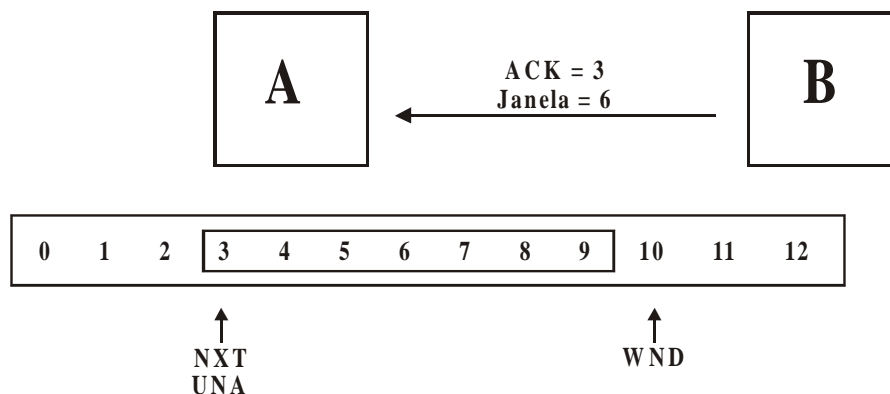
Figura 8 – Resultado de uma atualização do tamanho da janela.

Na Figura 8, a máquina B responde para A indicando que recebeu os bytes 1 e 2. Uma das particularidades do TCP é que anexado à informação do conhecimento do envio (ACK), B envia para A o tamanho do seu campo janela, para que A possa aumentar ou diminuir do tamanho da sua janela. O cálculo da janela de A continua a ser ACK+WND, mas como B enviou o novo tamanho de janela (WND) que agora é de 6, A pode aumentar o seu campo janela para 9, sendo 3 (ACK de recebimento) mais 6 (WND, campo janela recebido).

O campo de janela também poderia ser diminuído por B, fazendo A diminuir o ritmo de envio de segmentos para B. Este processo é contínuo durante uma transmissão TCP, levando ao controle de fluxo.

É possível, ainda, que qualquer dos dois lados, A ou B, envie um segmento contendo informações urgentes. Este segmento pode ser enviado por quaisquer dos lados, mesmo quando o seu campo janela está fechado (lotado). O *bit* de urgência é ativado fazendo com que o segmento seja enviado. O lado que recebe um segmento com o *bit* de urgência ativado, imediatamente processa os dados que estão no segmento que os contém.

O TCP não foi projetado para usar informações de não recebimento dos dados, por isso foi implementado uma forma para que fosse feito o reenvio das informações que por ventura viessem a se perder ou levassem muito tempo sem confirmação de recebimento. O método criado para tratamento do não recebimento dos dados foi a

estipulação de tempo máximo de espera de confirmação. O tempo de espera não é fixo; é calculado baseado nas primeiras informações de conhecimento de recebimento, tempo de processamento e envio de uma resposta e o tempo que leva em média para que o novo segmento chegue novamente na máquina emissora. Este é um cálculo complexo que envolveu algumas mudanças descritas no RFC 1122.

Segmento TCP

O segmento TCP está dividido conforme mostrado na Figura 9.

Figura 9 – Segmento TCP (PDU).

O campo *Source Port* – Porta Fonte – indica a aplicação que está disparando a conexão.

O campo *Destination Port* – Porta de Destino – indica a aplicação onde serão requisitadas informações no destino.

O campo *Sequence Number* – Número de Seqüência – contém o número de seqüência do primeiro octeto no campo de dados do usuário. O seu valor especifica a posição para o envio das próximas informações ao receptor. Este campo também indica

o número inicial da seqüência para o outro lado da conexão, para que os próximos números estejam baseados neste primeiro número enviado.

O campo *Acknowledgment Number* – Número de Conhecimento – é usado para confirmar o recebimento de dados recebidos. Seu valor é o próximo número de seqüência esperado pelo receptor que deve ser enviado pelo emissor. O Número de Conhecimento pode indicar o recebimento de mais de um segmento de dados. Para isto é enviado o número de dados recebido mais um.

O campo *Data Offset* – Deslocamento de Dados – indica onde começa a parte de dados (informações propriamente ditas) no segmento TCP. Separa cabeçalho e dados.

O campo *Reserved* – Reservado – é reservado para uso futuro de correções de eventuais erros do protocolo.

O campo URG indica que o segmento contém uma mensagem de urgência.

O campo ACK indica que o segmento contém uma mensagem de confirmação de recebimento.

O campo PSH indica que os dados no *buffer* de recebimento têm que ser enviados para a aplicação.

O campo RST indica que o segmento contém uma mensagem de reinício de conexão.

O campo SYN indica que o número de seqüência deve ser sincronizado. É usado na seqüência de conexão (*three-way handshaking*[17]).

O campo FIN indica que o segmento contém uma mensagem de finalização de conexão. A máquina que envia o pacote FIN não deseja receber mais dados.

O campo *Window* – Janela – indica quantos octetos é suportado pelo receptor. É complemento do campo de Conhecimento para cálculo da janela de envio.

O campo *Checksum* – Soma de Verificação – é usado para soma de verificação do segmento, garantindo que as informações chegaram livres de problemas no receptor. Verifica tanto cabeçalho quanto dados.

O campo *Urgent Pointer* – Ponteiro de Urgência – só é utilizado e verificado quando o *bit* de urgência está indicado. Serve para apontar onde começa a informação de urgência dentro da parte de dados do segmento TCP. O protocolo TCP não

---

[17] *Three-way Handshaking* – Aperto de mão de três vias – É o método usado pelo TCP para estabelecer uma conexão entre dois pontos.

implementa nenhum algoritmo de urgência, as informações são repassadas para a aplicação, que faz a solicitação de urgência e a aplicação receptora.

O campo *Options* – Opções – é utilizado para alguma eventual opção implementada por algum fabricante ou mudança no protocolo.

O campo Padding – Preenchimento – é utilizado para o preenchimento do cabeçalho TCP até um múltiplo de 32 *bits*.

O campo *Data* – Dados – é utilizado para tráfego de dados entre as aplicações.

Gerenciamento de Conexões

O protocolo TCP é um protocolo voltado a estados, ou seja, ele faz uso de regras para troca de informações entre as máquinas e para definição do estado da conexão.

As operações de conexão do TCP são abertura, transferência de dados e fechamento dos dados.

A operação de abertura de conexão é mostrada na Figura 10.
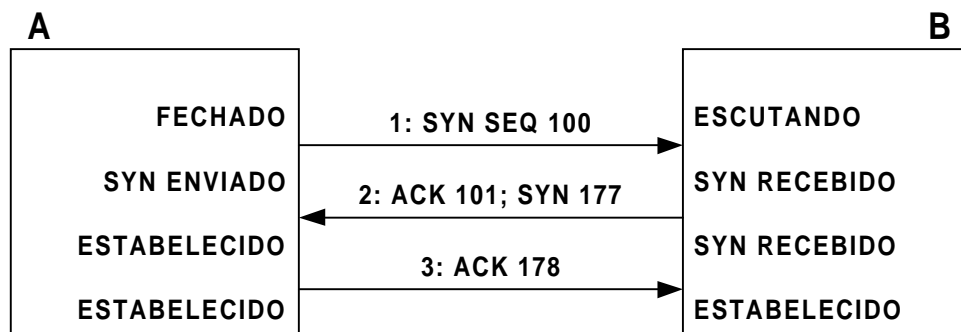
Figura 10 – Operações de abertura de conexão TCP.

Na Figura 10, A inicia o processo de conexão com B. A representa o lado com um *socket* ativo, enquanto B iniciou um *socket* passivo e espera conexões. A inicia o processo enviando para B um segmento com o bit de SYN ativado e o seu número inicial de seqüência (neste caso começa em 100). O número de seqüência pode ser qualquer número válido e não altera a conexão.

B envia de volta para A um segmento confirmando o recebimento do pacote inicial da transmissão. Este segmento tem o bit de SYN ativado e marcando o número de seqüência de B (neste caso 177) mais o bit de ACK ativado e marcando o número

que corresponde ao próximo número esperado para a seqüência de A (confirmando o recebimento de um segmento com a seqüência em 100).

Por último, A envia a confirmação do recebimento da seqüência de B, contendo ACK ativado e em 178 (SEQ+1) e tem, desta vez, o SYN desativado, confirmando assim, o fim do sincronismo de conexão.

Este processo que envolve os três passos descritos anteriormente é chamado de aperto de mão em três vias (*three-way handshaking*).

Está estabelecida, então, a conexão entre A e B. Neste mesmo processo, o TCP já se encarrega de calcular alguns tempos para uso futuro nas operações de limite de tempo.

Posterior as operações de abertura de conexão, segue a transferência de dados, mostrado na Figura 11.
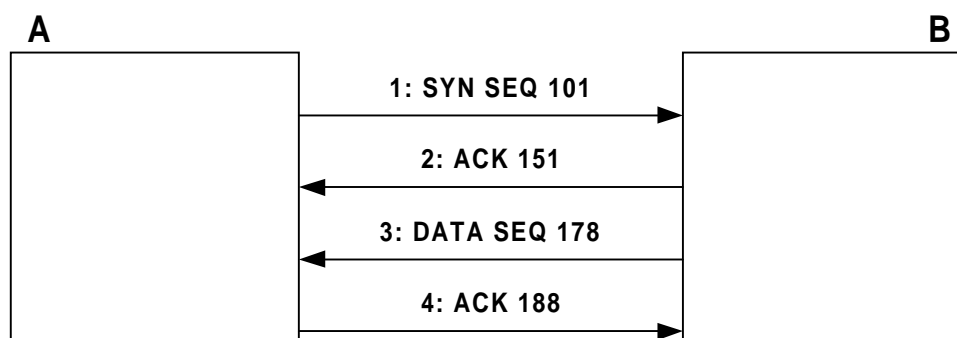
Figura 11 – Operação de Transferência de dados.

Na Figura 11, A inicialmente manda para B 50 *bytes*[18] com o segmento contendo o número de seqüência de 101 (estipulado no processo de abertura de conexão) e os *bits* de SYN e ACK não estão ativados. B recebe o segmento de A e envia um segmento de confirmação com o *bit* de ACK ativado e marcando 151, indicando o próximo número de seqüência esperado. Para a confirmação, B soma o número de seqüência que A enviou com os *bytes* recebidos, totalizando 151.

Posteriormente, B envia para A 10 *bytes*. Como o seu número de seqüência está em 178, B envia o segmento contendo os 10 *bytes* e com o número de seqüência em 178. A confirma o recebimento à B, com o segmento tendo o *bit* de ACK ativado e em 188 (178 + 10 *bytes*). A transmissão pode continuar envolvendo todos os processos de

[18] Será usada a medida em *bytes* para este exemplo hipotético.

cálculo de tempo para expiração, tamanho de janela e eventuais erros e consertos no envio dos dados.

Finalizando, é feito o fechamento da conexão entre as duas máquinas, mostrado na Figura 12.
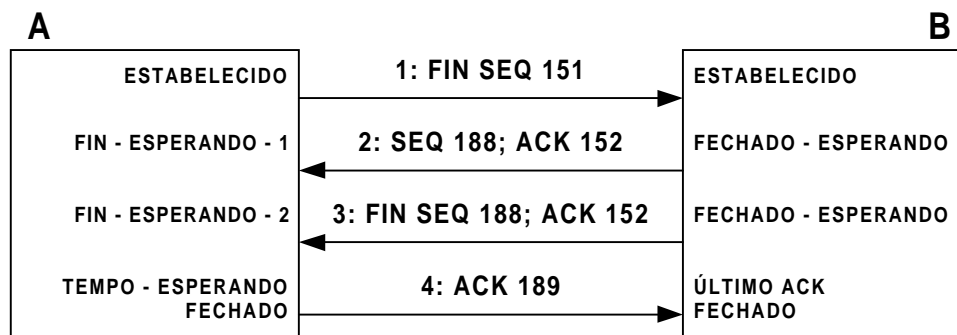
Figura 12 – Operação de fechamento de conexão TCP.

Na Figura 12, A envia para B um segmento solicitando a finalização da conexão. Este segmento tem o *bit* de FIN ativado e o número de seqüência em 151 (continuação da operação entre eles).

Caso a aplicação de B confirme que será encerrada a conexão, é enviado para A um segmento ratificando o processo de encerramento da conexão. Neste caso, B decide encerrar a conexão. No segmento de confirmação enviado para A, o *bit* de FIN está ativado e contendo 188 e mais uma vez é confirmado o recebimento do segmento de A para finalização (ACK em 152).

A confirma para B o recebimento, enviando um segmento com o *bit* de ACK ativado e em 189. Está encerrada a conexão entre A e B.

Tanto A como B poderiam estar enviando e recebendo dados em conexões com várias outras máquinas (B tem um *socket* passivo, provavelmente é um servidor), é necessário que haja algum modo para que ambos gerenciem estas conexões. Para isto foi criada a tabela de conexões, mostrada na Figura 13.

	Estado da Conexão	Endereço Local	Porta Local	Endereço Remoto	Porta Remota
Conexão 1					
Conexão 2					
Conexão 3					
Conexão n					

Figura 13 – Tabela de Conexões.

A tabela de conexões contém as informações de todas as conexões efetuadas na máquina e os seus estados (fechado, ouvindo, fechando, etc.). Para cada conexão é criada uma linha contendo 5 campos preenchidos conforme as seguintes informações:

A coluna de Estado da Conexão contém o estado que se encontra a conexão.

A coluna de Endereço Local contém o endereço IP da máquina local que respondeu à conexão. Quando o estado é *escutando*, o valor deve ser 0.0.0.0.

A coluna de Porta Local contém a porta local da conexão.

A coluna de Endereço Remoto contém o endereço IP da máquina remota da conexão.

A coluna de Porta Remota contém a porta remota da conexão.

2.4. UDP

2.4.1. Conceitos

O conjunto de protocolos da internet também abrange um protocolo de transporte sem conexão. Este protocolo é chamado de UDP (*User Datagram Protocol*), do inglês, Protocolo de Datagramas de Usuários. Este protocolo serve para pequenas transferências, onde não é necessária uma conexão ou que o tempo e custo operacional de conexão e desconexão seria muito alto. Este protocolo por não utilizar controle e fluxo de conexão é utilizado somente para envio de requisições e respostas a pequenos protocolos das camadas superiores.

O UDP, como o TCP, também utiliza o conceito de portas de comunicação, tendo sua estrutura mostrada na Figura 14.

32 bits	
Porta de Origem	Porta de Destino
Tamanho	Soma de Verificação
Dados	

Figura 14 – Formato do datagrama UDP.

A Porta de Origem identifica a porta de onde saíram os dados com destino ao receptor. Este item é opcional e quando não preenchido é inserido o valor padrão zero.

A Porta de Destino identifica a porta destino para onde estão seguindo os dados.

O Tamanho indica o tamanho do datagrama UDP incluindo o cabeçalho UDP e a parte de dados.

A Soma de Verificação é um valor opcional de 16 bits para verificação do cabeçalho e dados do UDP. A Soma de Verificação do UDP serve como complemento para a verificação do pacote IP.

O protocolo UDP, por ser simples, é utilizado para serviços que não exigem complexidade de transferências, controle de fluxo ou erros. Existem serviços que, por padrão, escutam portas UDP específicas (ANEXO B).

3. Segurança

3.1. Firewall

O *firewall* é a primeira solução proposta por profissionais de segurança de redes e computadores para o início da implantação das políticas de seguranças definidas numa empresa ou simplesmente a primeira opção para o início da própria política. O *firewall* é o conjunto de componentes (*hardware*, *software*) que restringem o acesso entre uma rede protegida e a internet ou outros conjuntos de rede (Chapman e Zwicky, 1995). Nestes componentes estão inseridas também as regras definidas em reuniões com gerentes para discussão de políticas de segurança, estrutura física da rede da instituição e qualquer outra definição ou decisão que tenha influência na segurança das máquinas em rede.

Os componentes principais no paradigma de proteção de redes por *firewalls* são: [1] as regras nas quais o tráfego de rede é submetido, [2] rotas para acesso às redes, [3] arquitetura física das redes e do próprio *firewall*, [4] as máquinas e [5] os roteadores.

3.1.1. Filtro de Pacotes

O Sistema de Filtro de Pacotes faz a passagem de pacotes vindos de uma rede externa (geralmente a internet) para a rede interna de maneira seletiva, ou seja, conforme definições anteriormente ajustadas.

Este trabalho é comum em roteadores que são chamados de *screening routers* que do inglês significa roteadores com telas. Para cada tipo de conexão efetuada é criada uma tela (janela) para que passem (sejam roteados) os pacotes de uma rede a outra; havendo alguma regra que proíba a passagem do pacote para a rede interna ou para a rede externa (internet) não é aberta a tela e o pacote é descartado no roteador. Este tipo de serviço é chamado de filtragem de pacotes.
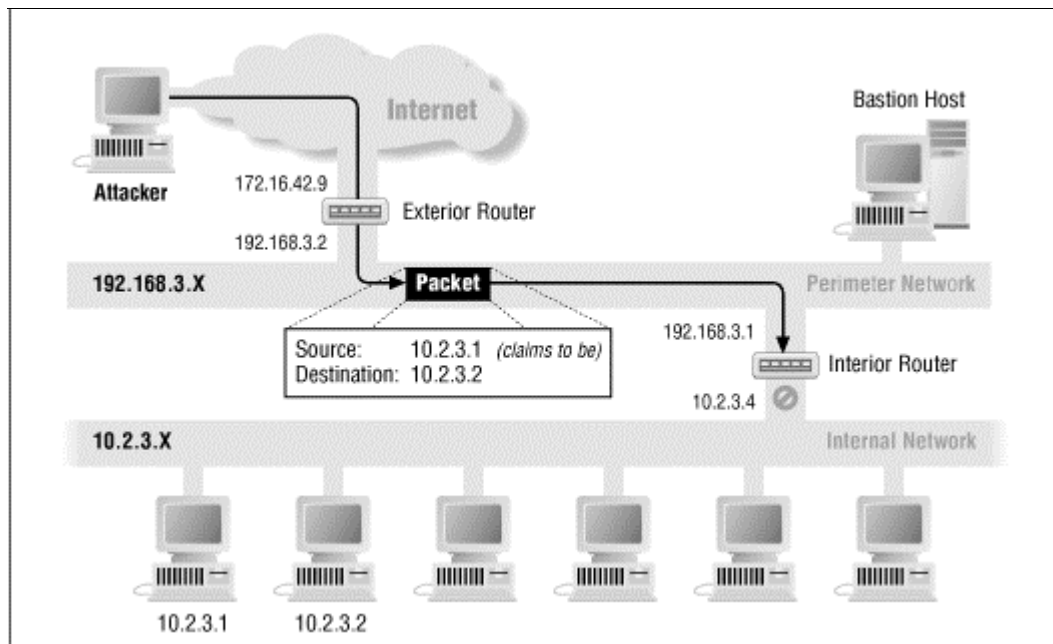
Figura 15 – Uso de *Screening Router* para filtro de pacotes.

Existem alguns parâmetros básicos para a configuração da filtragem de pacotes, sendo eles os seguintes:

- Endereço de origem do pacote: a permissão é baseada no endereço IP da máquina ou da rede emissora do pacote entrante.
- Endereço de destino do pacote: a permissão é baseada no endereço IP da máquina ou da rede receptora do pacote.
- Protocolo: configurando os protocolos que são permitidos passar pelo *screening router*.
- Porta de origem: definindo a permissão baseada na porta de origem nos protocolos UDP ou TCP.
- Porta de destino: definindo a permissão baseada na porta de destino nos protocolos UDP ou TCP.
- Interface de chegada: configurando por onde os pacotes podem entrar (interfaces de rede).
- Interface de saída: configurando por onde os pacotes podem sair (interfaces de rede).
- *Flags* TCP: a permissão é baseada nos *bits* de bandeira do TCP durante a conexão.

As permissões são configuradas utilizando a combinação das diversas regras de pacotes. Na Figura 16 é mostrado um exemplo de uma regra de filtragem em um roteador Linux (utilizando IPTABLES[19]).

```
# iptables -A FORWARD -i hdlc0 -o eth0 -s 200.154.25.6 -d 200.223.85.3 -p tcp
--sport 1024 --dport 23 -state --state INVALID -j DROP
```

Figura 16 – Exemplo de Regra de Filtragem utilizando IPTABLES.

3.1.2. Tradução de Endereços de Rede (NAT)

Geralmente, o *firewall* tem como finalidade proteger uma rede contra outras, externas ao ambiente. Comumente, utiliza-se para proteção de uma rede interna com classes de endereços IP reservados contra a rede mundial de computadores: Internet. Como os endereços internos são reservados e utilizados também por outras instituições no mundo, é necessário que, a partir de um certo ponto da rota do pacote IP, sejam trocadas as informações de origem para que o receptor consiga traçar uma rota de retorno ao emissor, quando de redes distintas.

O *Network Address Translation* (NAT), do inglês – Tradução de Endereços de Rede – é o serviço de mudança do endereço de origem ou destino (tradução).

O serviço de NAT possibilita que seja configurada uma tabela no roteador da rede, traduzindo os endereços de origem de pacotes que saem e de destino dos pacotes que entram. O processo é feito no momento em que uma máquina da rede interna solicita uma conexão a um servidor externo, passando pelo roteador. É feita a entrada na tabela de tradução de endereços dos IPs e portas de origem e destino, guardando as informações originais. Após isto, o roteador tenta fazer a conexão ao servidor modificando o endereço e porta de origem para o seu endereço de IP válido na rede real e uma porta livre; quando a resposta é recebida, o roteador usa a tabela de tradução para alterar o endereço e porta de destino para o endereço e porta da máquina que originou a requisição fazendo a entrega. A entrada na tabela de tradução de endereços é mantida até o término da conexão.

[19] Programa capaz de fazer filtro de pacotes em um roteador baseado em Linux. Regras são criadas a partir do IPTABLES para configurar o cerne do Sistema Operacional, permitindo ou não a passagem de pacotes.

	Endereço de Origem	Porta de Origem	Endereço de Destino	Porta de Destino	Porta Local
Conexão 1					
Conexão 2					
...					
Conexão n					

Figura 17 – Tabela de tradução de endereços.

Mascarar (*Masquerade*)

Quando o serviço de tradução de endereços é utilizado para prover acesso de uma rede falsa à rede mundial de computadores, onde os endereços IP são endereços reais e conhecidos pelos roteadores mundiais, é necessário o uso de máscaras de endereços, ou seja, o *masquerade* é utilizado para soluções de problemas onde uma rede de endereço falso e reservado troque informações com uma outra máquina externa com endereço real e válido.

O serviço de máscaras é necessário porque na internet não existem rotas para endereços reservados de uso em redes internas. A comunicação dos endereços de uma rede interna com a internet é feita através do roteador que liga as duas redes, porém o simples repasse dos pacotes de uma rede a outra causaria o problema no receptor, que está na rede mundial, pois não saberia a quem responder (o endereço de origem é um endereço reservado). É necessário que o roteador que faz a ligação entre as redes faça a mudança do endereço de origem para o seu próprio válido na internet e conhecido. É utilizada a tabela de tradução de endereços para o conhecimento do roteador dos tráfegos entre máquinas e a mudança (tradução) dos endereços de origem e destino. Na Figura 18 é mostrado o processo das máscaras.
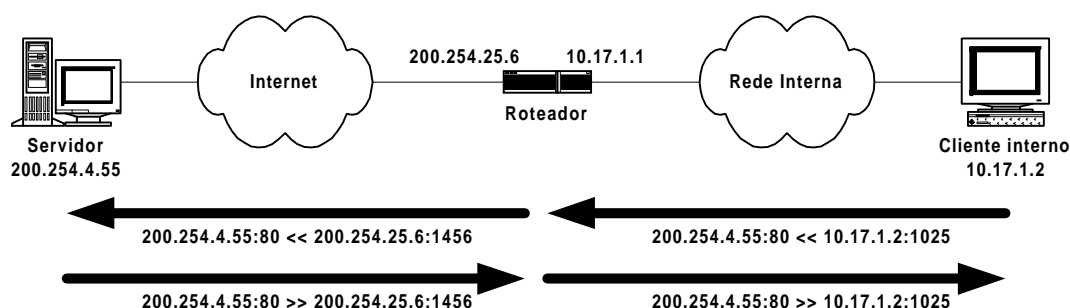
Figura 18 – Tradução de Endereços com máscara.

A máquina 10.17.1.2 requisita uma página HTML ao servidor de páginas 200.254.4.55. Entre eles, o roteador utiliza o seu endereço válido na internet (200.223.25.6) para modificar o endereço de origem da requisição e faz a entrada na tabela de tradução. O servidor de páginas responde ao endereço de origem (modificado para o roteador). O roteador recebe o pacote, verifica na sua tabela de tradução se existe uma entrada daquele endereço e, caso positivo, muda, desta vez, o endereço de destino para o endereço reservado e interno da máquina que fez a requisição (10.17.1.2).

Esta tradução é feita em todos os tipos de troca de informações entre qualquer máquina interna com qualquer máquina na internet, criando uma máscara entre a conexão dos dois pontos.

Tradução de Portas (PAT)

A Tradução de Portas é feita quando é necessário um re-direcionamento de pacotes e serviços de uma determinada máquina a outra. O caso mais comum de direcionamento de portas (serviços) é quando uma instituição possui um número limitado de endereços IP reais e válidos na rede mundial de computadores e precisa prover serviços onde é necessário este endereço com mais de uma máquina.

O *firewall* faz o re-direcionamento de serviços esperando conexões em portas específicas, modificando o endereço de destino para uma máquina interna com endereço falso e gravando estas informações na tabela de NAT para as respostas.

A Tradução de Portas também é utilizada para balanceamento de serviços, fazendo o direcionamento de portas conforme a quantidade de requisições, balanceando entre dois ou mais servidores.

O caso mais comum de utilização de tradução de portas é com finalidade de serviços de *Transparent Proxying* (Procuradores Transparentes), onde o *firewall* encaminha requisições em uma determinada porta para uma outra máquina, ou simplesmente à outra porta no servidor, onde existe um serviço de procuração para as conexões (neste caso sem o conhecimento do cliente, ou seja, transparente).

3.1.3. Serviço de Procuração (*Proxy*)

O serviço de procuração, *proxy* na língua inglesa, provê o acesso à internet para uma ou mais máquinas numa rede, por meio de um protocolo de comunicação no qual o servidor *proxy* faz a comunicação com o servidor real para o cliente.

O serviço de procuração é utilizado em soluções de proteção de redes, pois é facilmente configurado como o único ponto de comunicação com a internet, aumentando significativamente as defesas. O fator fundamental no uso do *proxy* para defesa é que não é necessária a configuração de defesas mais apuradas em cada estação que acessa a internet; é preciso apenas fazer tal modificação no único ponto de acesso: o servidor *proxy*.

O acesso a servidores na internet é feito por meio do procurador, funcionando transparentemente para o servidor real e o cliente interno, no qual o procurador está servindo. Por meio da instalação e configuração de um pequeno cliente na estação de trabalho interna, todas as requisições de determinado serviço são direcionadas ao servidor procurador, que se encarrega de requisitar o serviço ao servidor real e devolver ao cliente interno. Segundo Zwicky, Cooper e Chapman (2000, p.225), "O usuário tem a ilusão de lidar diretamente (ou quase diretamente) com o servidor na internet, com um mínimo de interação direta com o *dual-homed host*" (tradução nossa).
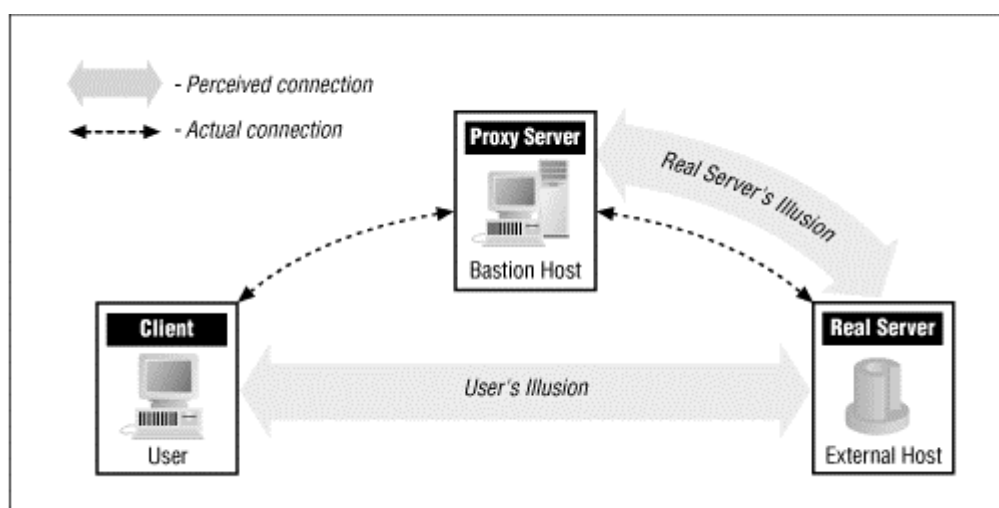
Figura 19 – Procuradores: realidades e ilusões.

A forma de funcionamento dos servidores *proxy* varia conforme os serviços. Alguns serviços provêem o serviço de procuração automaticamente; outros necessitam

mudanças no Sistema Operacional do cliente ou de um cliente específico instalado na estação cliente.

Proxy-Aware Application Software

Neste tipo de procurador, o software necessita saber como conectar ao servidor de *proxy*, ao invés de saber como conectar ao servidor real. Alguns exemplos deste tipo de configuração são o serviço de procuração *Igateway* e os navegadores internet *Netscape Navigator* e *Internet Explorer*.

O *Igateway* é um cliente para serviços de procuração dos protocolos *FTP* e *Telnet* que apenas funcionam em máquinas *Sun*[20].

Os navegadores para páginas internet *Netscape Navigator* e *Internet Explorer* possuem clientes de procuração para os protocolos *FTP*, *HTTP*, *HTTPS*, *SOCKS* e *Gopher*.

Proxy-Aware Operating System Software

Neste tipo de procurador, o Sistema Operacional da estação de trabalho é modificado, fazendo com que os serviços de *proxy* sejam transparentes aos programas. O Sistema Operacional checa se uma requisição deve ser enviada ao servidor *proxy*.

Proxy-Aware User Procedures

Neste tipo de procurador, o usuário usa o software cliente, que não sabe ou não tem opção de configuração de serviços de procuração, para conectar ao servidor de procuração e solicitar ao servidor de procuração que conecte ao servidor real.

O usuário deve seguir procedimentos para que o serviço de procuração funcione, por exemplo, no serviço de procuração do protocolo FTP: o usuário utiliza um cliente comum de *FTP*, porém faz a conexão ao servidor de procuração ao invés de conectar-se ao servidor real. Quando conectado no servidor de procuração, o usuário especifica, além do nome e senha, o servidor real que deseja conectar.

---

[20] *Sun* – Sistema Operacional UNIX (SunOS) criado pela *Sun Microsystems*.

Proxy-Aware Router

Neste tipo de procurador, o serviço de procuração é totalmente transparente ao usuário, Sistema Operacional e software cliente. As conexões são desviadas ao procurador pelo roteador entre a estação cliente e o servidor real. O roteador tem que possuir este tipo de funcionalidade para que seja possível este tipo de serviço, podendo também fazer ele mesmo o serviço de procuração.

Este tipo de procuração é chamada de híbrida, pois é necessário o uso de filtro de pacotes ou serviços de *proxy* transparente.

3.1.4. Arquiteturas

A configuração da forma de proteção por um *firewall* é variável conforme a necessidade e composição na qual a rede da instituição está implantada. Para cada tipo de rede é necessária a configuração de um tipo de *firewall.* Existem conceitos padrão para a construção da arquitetura de proteção de rede, porém as regras sempre são mudadas e instituídas conforme a política de segurança de instituição. Atualmente, existem algumas variações nas arquiteturas principais de *firewalls*, onde estão se constituindo novos paradigmas e padrões.

As arquiteturas principais dos *firewalls* são: [1] *Dual Homed Host*; [2] *Screened Host*; [3] *Screened Subnet*.

Dual Homed Host

Do inglês, Máquina de Duas Casas, é composto de um único sistema com pelo menos dois dispositivos de rede. Este sistema normalmente é configurado de tal forma que os pacotes não são diretamente passados de uma rede para outra. As máquinas da internet podem se comunicar com o *firewall* da mesma maneira que as máquinas da rede privada, porém o tráfego entre as duas redes é controlado pelo *firewall*.
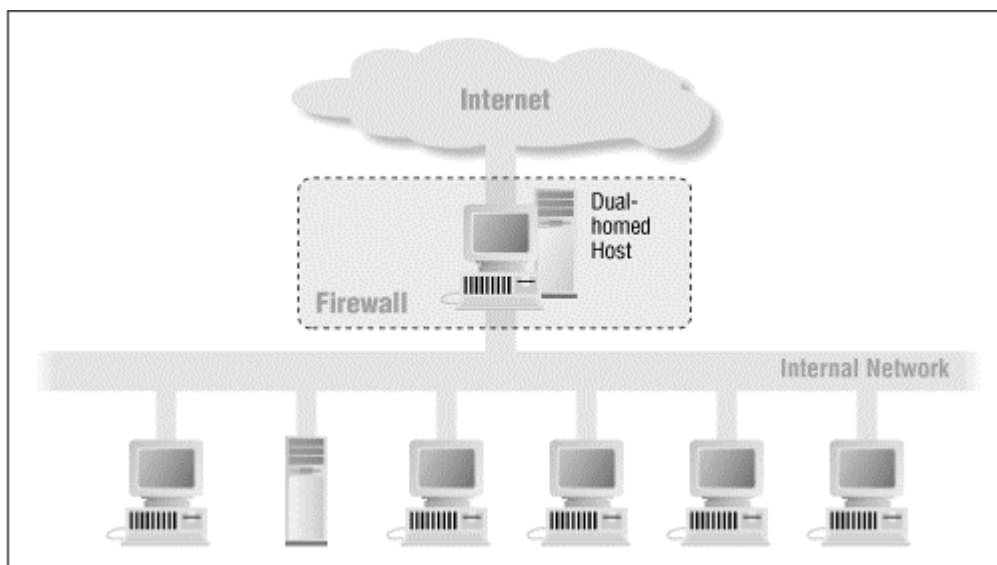
Figura 20 – Arquitetura de *firewall Dual Homed Host*.

Neste tipo de arquitetura, o *firewall* pode agir como o roteador entre estas redes, porém; na arquitetura chamada de *Dual Homed Host*, a funcionalidade de roteador é desabilitada, assim como a comunicação entre redes (roteamento entre redes). Para a comunicação das máquinas internas para máquinas externas (internet) são utilizados serviços de *Proxy*.
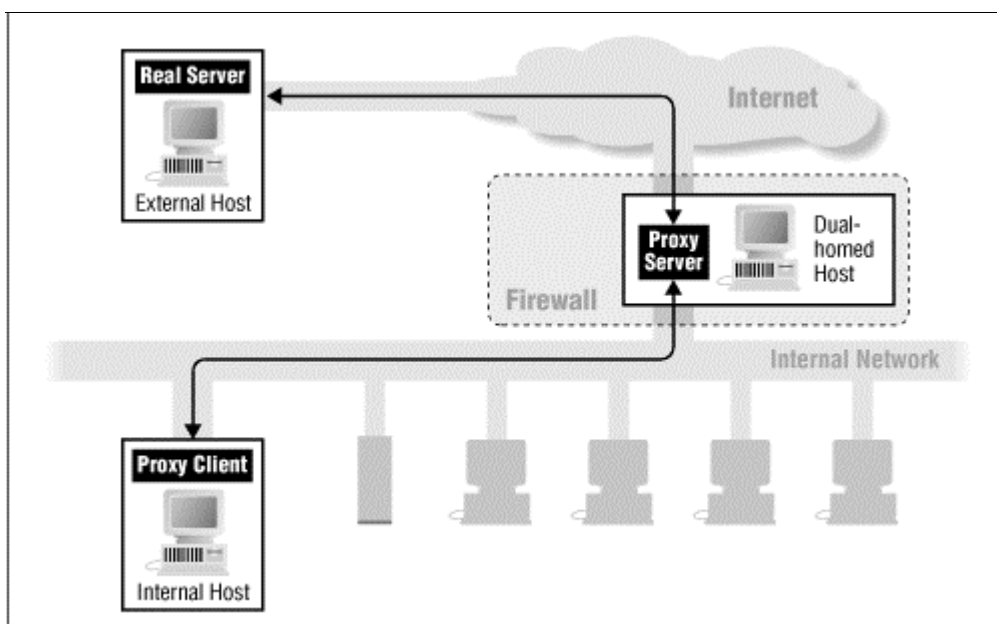
Figura 21 – Arquitetura de *firewall Dual Homed Host* com comunicação via *Proxy*.

Screened Host

A arquitetura *Screened Host*, assim como na arquitetura *Dual Homed Host*, disponibiliza os serviços de proteção com uma máquina ligada a duas redes, porém os serviços de rede são disponibilizados por uma máquina ligada a rede interna. Neste tipo de arquitetura, a segurança é feita através de filtro de pacotes no roteador que liga as duas redes.

A máquina interna responsável pelos serviços de rede é chamada de *bastion host*[21].
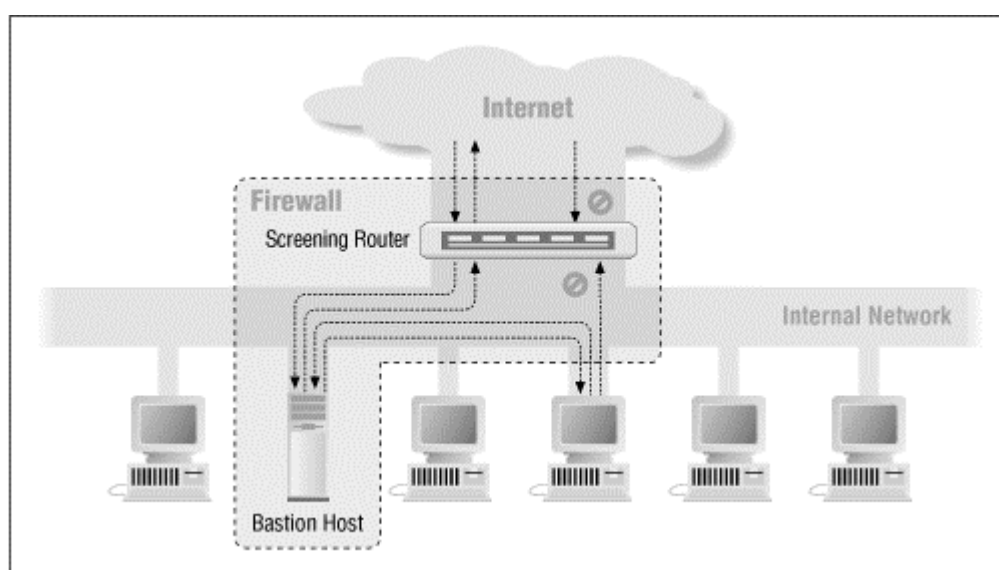
Figura 22 – Arquitetura *Screened Host***.**

O roteador (*Screening Router*) é configurado para que a única máquina que receba conexões e requisições externas seja o *Bastion Host*; porém, mesmo sendo a única máquina que recebe conexões, o roteador é configurado de modo a permitir somente os serviços definidos (filtrando os demais). Devido ao perigo eminente que o *Bastion Host* é submetido, a definição de segurança desta máquina é total.

O roteador é também responsável pela permissão da conexão da rede interna à rede externa diretamente ou indiretamente. É possível haver 3 tipos de permissões: [1] não é permitido nenhum tipo de conexão direta a rede externa, apenas por meio de Servidores *Proxy*; [2] é permitido o acesso de máquinas internas à rede externa, com filtro de pacotes e serviços; [3] é apenas permitido o acesso à rede externa para somente

[21] *Bastion Host* é qualquer máquina responsável por um tipo de serviço a máquinas numa rede externa.

algumas máquinas definidas pela política de segurança interna, deixando a conexão das demais máquinas por meio de Servidores *Proxy* ou não permitindo o acesso a rede externa.

Screened Subnet

A arquitetura de *firewall Screened Subnet* adiciona mais segurança à arquitetura *Screened Host* retirando o *bastion host* da rede interna, colocando-o numa rede periférica e isolando esta rede periférica da rede interna.

A *Screened Subnet* é considerada como o último nível de arquitetura em *firewalls*, isolando totalmente os *bastion hosts*, pontos que podem ser atacados diretamente. O impacto na segurança é significativo, apesar do aumento na complexidade de rotas e configuração de filtros e regras, pois são necessários dois roteadores com filtragem de pacotes (um roteador externo e outro interno) e uma nova rede separada da rede corporativa interna.
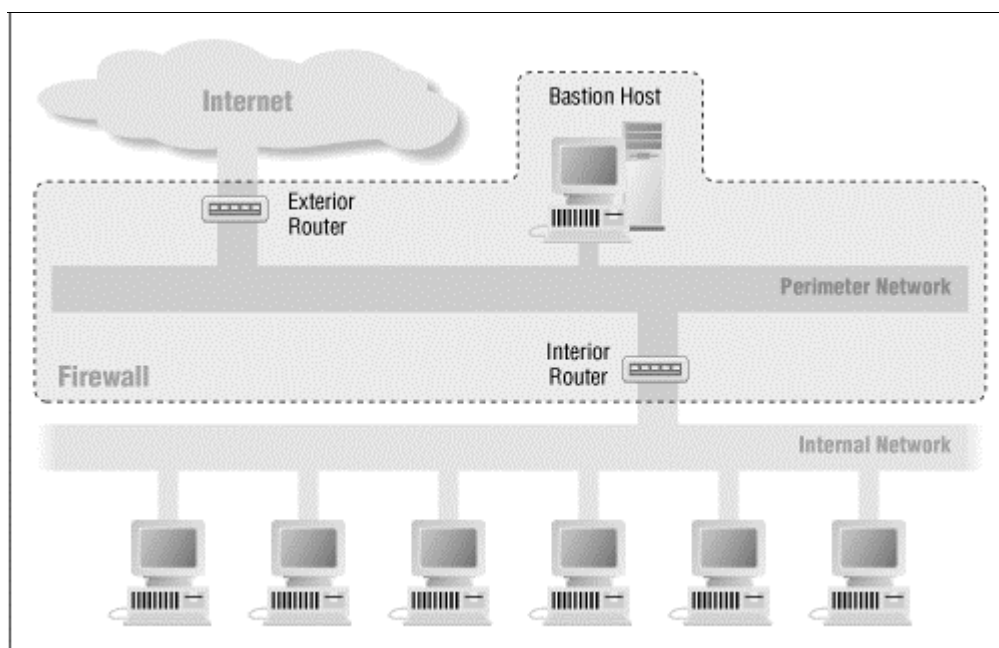
Figura 23 – Arquitetura *Screened Subnet.*

A segurança nesta arquitetura é maior, pois, ao contrário das arquiteturas anteriores e suas variações, quando um *bastion host* sofre uma invasão, a rede interna não está vulnerável, necessitando que ainda seja quebrada a segurança no segundo

roteador. O nível de complexidade também aumenta para o invasor que não tem conhecimento dos endereços falsos da rede interna, nem sequer a rota para tais máquinas.

A configuração da filtragem de pacotes nos dois roteadores está separada, podendo ter, no roteador interno, regras mais rigorosas de filtragem e negação de serviços, enquanto são inviáveis certos tipos de filtragem de pacotes no roteador externo. Alguns exemplos de filtragem que podem ser implantados nos roteadores internos são: [1] criação de regras para permissão de conexões para somente serviços em servidores *proxy*, fazendo com que as máquinas internas só possam utilizar serviços internet através de procuradores; [2] criação de regras para permissão de abertura de conexões da rede interna para externa, somente neste sentido, ou seja, somente a rede interna poderá abrir conexões, evitando conexões a *back-doors*[22], máquinas infectadas por vírus ou com programas de precedência duvidosa.

O uso da rede periférica é vantajoso porque, em invasões a *bastion hosts*, não é possível fazer monitoramento da rede interna, muito menos "escutar" o seu tráfego, impossibilitando o "roubo" de informações sigilosas como senhas de acesso e contas de usuários.

3.1.5. Políticas e Regras

Para a criação de um ambiente seguro é preciso tomar certo cuidado sobre alguns pontos que não são necessariamente softwares ou dispositivos. O cuidado sobre segurança deve ser uma filosofia de funcionamento dentro das instituições e estar em nível mais alto do que meros programas.

Privilégio Mínimo

O princípio fundamental da segurança é o principio do privilégio mínimo. Quanto menos privilégios e direitos tiverem os funcionários, softwares ou qualquer coisa relacionada, mais seguro estará o ambiente.

O princípio do privilégio mínimo significa dar ao objeto (máquinas, usuários, etc) os direitos apenas necessários para o seu funcionamento ou trabalho. Não é

---

[22] *Back-door* – Porta de Trás – Serviços que funcionam numa máquina permitindo a conexão de pessoas externas sem qualquer tipo de verificação e com super privilégios.

necessário que os usuários tenham acesso de gravação (ou até mesmo leitura) aos arquivos de configuração de um sistema. É preciso definir meios para que sejam delegadas capacidades de acesso restritas ao objeto; se um funcionário é encarregado pelo *backup* da empresa, é delegado acesso de somente leitura aos arquivos que devem ser guardados e gravação no dispositivo de gravação do *backup*, se possível, somente por meio de um programa e em horários definidos.

Defesa em Profundidade

Para que as defesas sejam efetivamente asseguradas são necessárias algumas precauções sobre níveis de segurança, redundância e educação de pessoal.

Num ambiente seguro é necessário que sejam tomadas medidas como *backups* periódicos, redundância de *firewalls*, *firewalls* em níveis internos, educação dos funcionários sobre segurança, segurança individual das máquinas, etc.

É muito importante a profundidade das defesas, pois em momentos de falhas ou quebras de segurança, as dificuldades de ataque para pessoas mal intencionadas ainda continuam altas.

Ponto de Estrangulamento (*Choke Point*)

O ponto de estrangulamento, do inglês *choke point*, é o ponto onde todo o tráfego de uma determinada rede passa. É uma política de segurança bastante comum, onde é obrigatório que o tráfego passe por um ponto e a segurança é focalizada neste ponto.

Numa defesa com *firewalls*, o ponto de estrangulamento é justamente os screening routers, redes periféricas e todos os componentes do *firewall*.

Ponto Mais Fraco

O ponto mais forte na defesa de rede é justamente o ponto mais fraco da rede, ou dispositivos que a defendem.

Os atacantes sempre irão procurar qual é o serviço mais vulnerável para que possam efetuar as tentativas de ataque e invasão, portanto, é de muita importância que os pontos mais fracos na defesa de redes sejam monitorados quando não possam ser desativados.

Num ambiente de defesa com *firewalls*, depois dos *bastion hosts*, os pontos fracos do *choke point* são os mais atacados.

Falha Segura

É necessário que sejam previstos os casos de falhas, em quaisquer níveis, para que as falhas sejam tratadas de forma segura.

As falhas devem ser tratadas de forma que cortem todo o acesso aos sistemas, recursos e tudo mais que seja importante para o ambiente de rede. É importante que seja entendido que, em caso de falhas e corte de acesso aos recursos aos usuários externos, é também cortado o acesso para os usuários internos; o sistema está em estado de falha!

Definindo as regras padrão a serem tomadas quando em estado de falhas, ficam duas possibilidades:

- Tudo que não é explicitamente permitido é proibido;
- Tudo que não é explicitamente proibido é permitido.

Participação Universal

É importante que todos os participantes da rede (funcionários numa instituição, por exemplo) estejam de acordo com as políticas e participem das responsabilidades necessárias ao funcionamento.

Em nada adianta as políticas de segurança com *firewalls* bem configurados e ambientes seguros, se um funcionário tem acesso a redes externas por meio de conexões discadas ou formas semelhantes.

A participação deve ser voluntária e em casos de não participação, é necessário que sejam tomadas medidas para a participação. A melhor dela deve ser a que leve à participação voluntária.

Diversidade de Defesa

Para que um ambiente esteja realmente seguro é necessário que haja, em pontos estratégicos, o uso de defesas com redundância ou programas diferentes.

A diversidade na defesa é importante, pois nenhum programa ou dispositivo é infalível e, o uso de mais de uma solução, acaba minimizando os problemas em casos de tentativas de ataques e invasões.

Um bom exemplo que justifica o uso de mais de uma política de defesa são os vírus de computadores. Geralmente os programas antivírus não acertam no diagnóstico de vírus em um determinado arquivo, porém um outro poderia acertar.

3.1.6. Segurança das Máquinas em Rede

A segurança dos meios de transmissão e dos serviços não é suficiente para manter a segurança da rede; é preciso que todas as máquinas que são possíveis alvos de ataques e de tentativas de invasão, como os *bastion hosts* e *screening routers,* sejam configuradas de modo a manter o maior nível de segurança possível.

Para a manutenção da segurança nas máquinas, é preciso que sejam seguidos alguns procedimentos básicos:

- Desligamento de todos os serviços que não são necessários na máquina. Um servidor de e-mails não precisa ter o serviço de HTTP funcionando, a menos que seja também servidor de páginas. A parada dos serviços desnecessários também aumenta a quantidade de memória livre e libera mais tempo de CPU.
- Os serviços TCP e UDP que estejam habilitados nos roteadores e são necessários devem ter liberado o acesso para apenas administradores e máquinas específicas.
- Os serviços remotos nos roteadores (*source routing*[23], configuração remota) devem ser desabilitados.
- As portas (serviços) de gerenciamento nos roteadores devem estar protegidas e as desnecessárias devem ser desabilitadas.
- As senhas de máquinas importantes na rede devem ter senhas duráveis.

> "Ter pelo menos oito caracteres de tamanho, não ser palavras, não começar com números, ter letras, números e incluir pelo menos um dos caracteres especiais (isto é ,./<>;':"[]\{}|~!@#$%^&*()_+`-=). As senhas devem ser alteradas em no máximo 90 dias.".(SNAC, 2001, p.9) (tradução nossa).

- Todo acesso a qualquer serviço deve ser registrado nos arquivos de registro do sistema.

---

[23] *Source Routing* – Rota pela Fonte – Serviço de definição de rotas baseadas na origem do pacote. Fazem parte do pacote "Rotas Avançadas" ao contrário dos serviços de rotas comuns que indicam apenas rotas de destino sem verificar a origem do pacote.

- As contas de usuários devem ser limitadas aos serviços para qual foram configuradas. Se o usuário tem uma conta de e-mail num servidor, sua conta deve apenas ter acesso aos serviços de SMTP e POP (IMAP em alguns casos). Se possível, limitar o tempo de ociosidade do usuário (tempo parado e conectado no serviço) e quando o usuário pode efetuar conexão (horários para conexão).

3.2. Sistema de Detecção de Intruso (IDS)

O processo de monitoração de eventos ocorridos num sistema de computadores ou na rede que engloba a procura de sinais de invasão, tentativas de corrupção da confiabilidade, integridade, disponibilidade ou transposição da segurança, é denominado como detecção de intruso. O Sistema de Detecção de Intruso é aquele que (em software ou hardware) faz este monitoramento, de forma automatizada, tomando decisões aos estímulos gerados pelos fatos ocorridos.

O Sistema de Detecção de Intruso é utilizado por instituições como forma complementar para proteção de redes de computadores, monitorando o comportamento da rede ou de uma simples máquina à procura de tentativas de invasão que tenham burlado ou passado por outros mecanismos de rede. No entanto, a sua instalação num ambiente de rede não garante a sua segurança, sendo necessário o uso dos meios comuns de proteção como *firewalls*, servidores de antivírus, etc.

3.2.1. Método de Detecção

Os IDS necessitam capturar e analisar os dados para então alertar sobre possíveis tentativas de ataque. Existem diversos tipos de captação dos dados para análise, porém os métodos de detecção para a análise são mais restritos e basicamente se dividem em dois grandes grupos.

Baseado em Comportamento

No tipo de detecção baseado no comportamento, o IDS faz um estudo sobre o comportamento da utilização de recursos, horário da utilização, tipo de aplicações usadas, etc, gerando um padrão de comportamento considerado normal. Os estudos vão

desde o uso de CPU, carga de rede e média de uso da memória a horários de conexão, tipo de aplicação para cada usuário e comportamento de utilização dos sistemas.

Quando definido um comportamento padrão para o ambiente testado, o IDS passa a comparar todo o uso dos recursos ao estudo prévio, decidindo o que é um mau uso ou ataque. Por exemplo, para um determinado usuário que possui o hábito de utilizar o sistema somente em horário comercial e executar aplicativos simples como leitores de e-mail e navegadores, entrar no sistema às quatro horas da manhã e compilar uma dezena de programas é um forte indício de uma invasão.

O uso da detecção baseada no comportamento pode gerar falsos alertas ou não detectar certos movimentos. A atualização anual de um servidor pode ser alertada como ataque pelo grande fluxo de arquivos em um determinado horário, quando não é comum o uso da máquina (horário adequado para a tarefa de atualização); ou um invasor que, conhecendo as configurações do IDS, faz o uso de ferramentas para que seu comportamento destrutivo seja visto como normal.

Baseado em Conhecimento

No tipo de detecção baseado no conhecimento, o IDS procura por assinaturas de ataques previamente configuradas, tal como programas de antivírus. O IDS procura por seqüência de ações não aceitáveis ou por seqüência de *bits* que caracterizam uma assinatura de ataque. As assinaturas de ataques necessitam ser constantemente atualizadas e um novo tipo de ataque pode passar pelo IDS pela falta de uma regra para a sua captura.

3.2.2. Arquitetura

A arquitetura de um IDS está ligada à forma como seus componentes funcionais encontram-se arranjados em relação uns aos outros. Os fatores que mais influenciam na arquitetura de um IDS são a localização e o alvo.

Segundo o Alvo

Na divisão da arquitetura do IDS segundo o alvo, o fator analisado é a fonte de dados que será trabalhada. Existem três tipos de IDS:

- Baseada na máquina (*host-based*): O IDS captura informações em um computador individual, fazendo uma análise profunda e precisa sobre o tipo de ataque, os estragos feitos, usuários envolvidos e processos comprometidos. Todavia, fica restrito a máquina que está instalado. São utilizados como fonte de pesquisa os arquivos de auditorias de sistemas e os arquivos de registros (syslog). As vantagens do IDS baseado na máquina são: [1] detecção de ataques que nenhuma outra arquitetura poderia detectar; [2] pode operar em ambientes de rede criptografados; [3] não há problemas onde as máquinas são interligadas com *switches.* As desvantagens: [1] são difíceis de monitorar, pois necessitam que seja configurada cada máquina monitorada; [2] por fazer parte da máquina atacada, pode ser atacado e desabilitado; [3] não detecta eventos de rede que não são diretamente direcionados a sua máquina; [4] por utilizar os arquivos de auditoria do Sistema Operacional, é necessário bastante espaço de disco para tais arquivos; [5] utilizam recursos computacionais da máquina que protegem.

- Baseada na rede (*network-based*): O IDS captura as informações diretamente da rede, fazendo a análise de todos os pacotes que trafegam em determinados protocolos. Podem funcionar de forma “invisível”, ou seja, sem endereço IP válido na rede que monitora e também pode ter mais de um dispositivo de rede (sensores). As vantagens do IDS baseado na rede são: [1] poucos IDSs podem monitorar uma rede de computadores ampla; [2] o impacto no funcionamento da rede é nulo ou quase nulo, pois é utilizada uma máquina exclusiva para o IDS; [3] podem ser altamente seguros contra ataques, pois podem ser configurados de forma “invisível”. As desvantagens são: [1] podem falhar e até mesmo não funcionar em ambientes com muito tráfego de rede ou que estão em velocidades altas, pois não acompanham a velocidade do tráfego de informações; [2] não funcionam em ambientes onde a rede é interligada por *switches*; [3] não pode operar em ambientes de rede criptografados; [4] IDS baseado na rede não consegue informar se o ataque foi ou não efetuado com sucesso.

- Baseada na aplicação (*application-based*): Particularidade do IDS baseado na máquina, o IDS baseado na aplicação monitora eventos entre softwares e usuários a procura de mau uso de sistemas. Geralmente utiliza arquivos de registros de transações (*transaction logs*) das máquinas à procura de invasões e ataques. As vantagens do IDS baseado na aplicação são: [1] pode monitorar atividades entre usuários e aplicações, verificando má atividade de certos usuários; [2] pode operar em ambientes criptografados, pois trabalham com arquivos de transação que estão em formato decriptado. As desvantagens são: [1] pode ser mais vulnerável do que o IDS baseado em máquina, pois os arquivos de transação de aplicações não são tão bem protegidos como arquivos de sistema; [2] somente detecta problemas de aplicações de usuários, sem habilidades de monitoramento de *Cavalos de Tróia*[24], ataques, invasões e similares.

Segundo Localização

O segundo aspecto que também é muito importante para a arquitetura do IDS é onde e como estarão distribuídos os componentes de um IDS. O módulo de captura, análise e alerta de um IDS podem estar todos separados em várias máquinas na rede ou todos reunidos somente em uma máquina. Para Bace e Mell (2001, p.10), “Estratégia de Controle descrevem como os elementos de um IDS são controlados e, ainda, como as entradas e saídas de um IDS são gerenciadas” (tradução nossa).

As estruturas centralizadas têm a vantagem de facilidade de instalação configuração, operação e ganho de desempenho. A simplicidade de manutenção e desenvolvimento garante vantagens em relação às demais arquiteturas, porém a complexidade dos sistemas atuais aliada à diversidade e dimensões da maioria das instalações computacionais, requer que sejam utilizadas soluções descentralizadas.

Os sistemas de proteção de redes de computadores requerem redundâncias para a tolerância à falhas ocasionando em maior segurança. Este fator leva a instalação de uma arquitetura descentralizada com módulos separados e independentes trocando informações entre si e garantindo esta redundância. O uso de tal arquitetura, no entanto, leva à complexidade na troca de informações entre os módulos, problemas como

[24] Pequenos softwares que realizam tarefas destrutivas no sistema como um todo.

algoritmos criptográficos, protocolos de autenticação, técnicas de detecção de falhas, assinatura digital e a própria troca de informações entre os componentes, elevando os problemas de controle do uso das redes e desempenho da aplicação.

Os sistemas híbridos são configurados de forma a fazer a interação entre sistemas com arquitetura centralizada e distribuída, tirando proveito das vantagens individuais de cada um e selecionando a melhor solução para cada sub-rede dentro de uma rede corporativa e ampla.

Bace e Mell (2001, p.12) dividem os IDS segundo a localização, em três formas: [1] controle centralizado; [2] parcialmente distribuído, onde são utilizados sistemas distribuídos e centralizados reportando hierarquicamente a um componente principal; [3] totalmente distribuído.

3.2.3. Comportamento Pós-detecção

Quando descoberta a informação do evento, o IDS necessita tomar uma decisão. O comportamento do IDS diante de um evento (variando entre ataques, tentativas, etc.) é conhecido como Resposta ao Evento e é dividido em resposta passiva ou resposta ativa.

Ativo

As respostas ativas a uma informação de uma ação são ações automatizadas exercidas pelo IDS na busca de coleta de informações adicionais sobre o ataque, mudança do ambiente de rede na qual o atacante tenta invadir, ações contra o invasor, dentre outras.

Os Sistemas de Detecção de Intruso mais “inteligentes” tomam uma série de ações no momento da invasão, onde são guardadas todas as informações sobre os passos tomados pelo atacante, estados dos sistemas e da rede; re-configuração de *firewalls* e roteadores para prevenção de novos ataques; envio de pacotes TCP para reinício de conexão (Capítulo 2, Seção 3) e ações contra o atacante, como varredura de portas e outras medidas hostis (este último bastante perigoso e desaconselhado por motivos de falhas na detecção, ilegalidade da ação e ataques a redes e usuários inocentes).

Passivo

As respostas passivas são mais comuns em IDS e finaliza o processo de captura, análise e alerta de intrusão. Utilizadas apenas para informação de intrusão, para ações posteriores de outros sistemas ou intervenção humana, as respostas passivas podem ser alarmes e notificações, em arquivos de sistema, e-mail, *pager*, *snmp traps*[25], etc.

[25] Eventos gerados para um servidor de gerência de rede utilizando o protocolo SNMP, protocolo desenvolvido para monitoramento e gerência de redes.

3.3. Ataques e Vulnerabilidades

Os ataques contra redes de computadores aumentam consideravelmente a cada ano, tornando cada vez mais necessário o uso de mecanismos de defesa, a fim de minimizar a ação de invasores e pessoas mal intencionadas. Segundo o CERT[26], até novembro do ano de 2001, foram cadastrados mais de 30.000 ataques contra redes e máquinas (Figura 23), indicando um aumento de quase 100% em relação ao ano anterior.
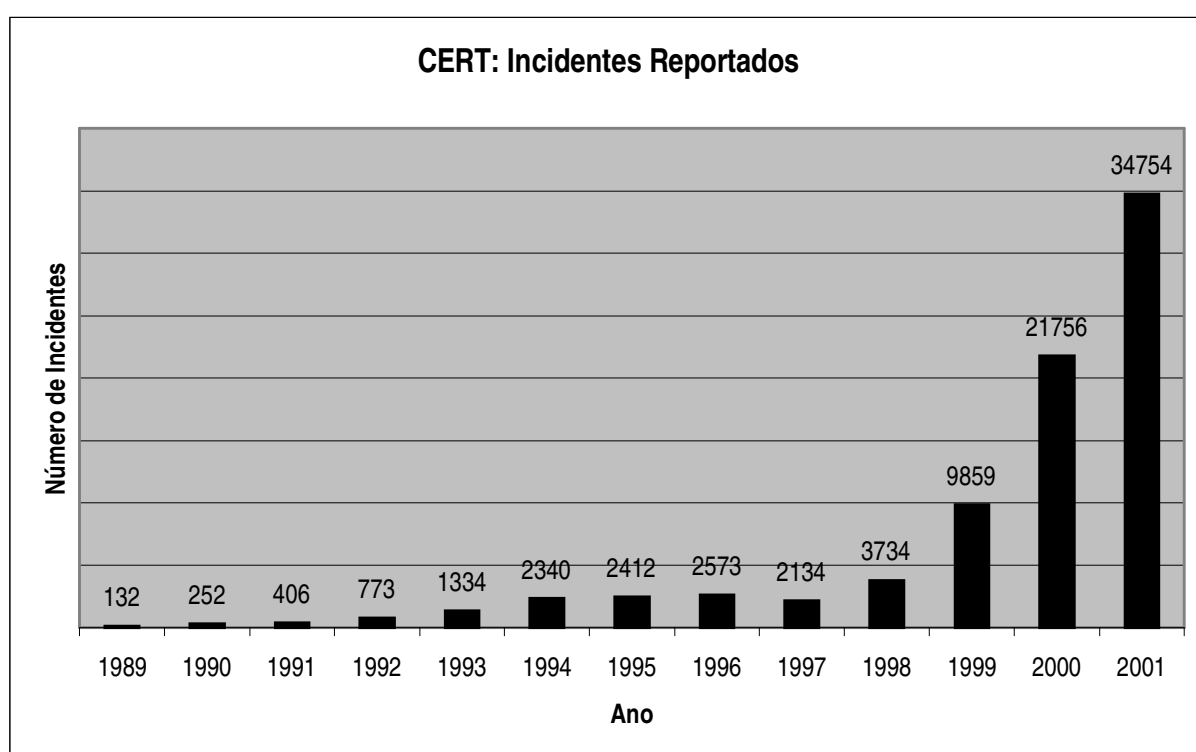
Figura 24 – Incidentes Reportados ao CERT nos últimos anos.

A grande maioria dos ataques registrados tem como causa a falta de cuidado dos administradores de rede, problemas de erros em aplicativos e nos Sistemas Operacionais e a má configuração dos serviços. Apenas alguns poucos ataques são resultados de uma exploração da fragilidade dos protocolos de rede ou um esforço mais inteligente de um

---

[26] CERT – *Computer Emergency Response Team* – http://www.cert.org. Centro de pesquisas sobre incidentes de segurança.

cracker[27], significando que a simples criação e implantação de políticas de segurança, aliado a uma boa conduta e administração dos sistemas e redes reduz consideravelmente os índices de ataques e invasões.

Outro fator importante que é visível no cenário atual é a banalização e facilitação do acesso a ferramentas de ataques – cada vez mais fáceis de operar, sem necessitar do conhecimento profundo pelo atacante – para uso indiscriminado que tem aumentado os problemas de ataques enfrentados pelas instituições. Estes tipos de ferramentas são denominados *script kiddies*, programas criados por *crackers* para explorar a fragilidade de Sistemas Operacionais e vulnerabilidade dos serviços ou sua má configuração.

A exploração de erros, uso de fragilidades e grande criatividade dos atacantes ocasionam em um grande número de ataques diferentes. Estes ataques geralmente estão baseados nas mesmas fragilidades e erros, mudando apenas o nome, alvo e intensidade do uso. A corrupção dos sistemas é ocorrida de maneira bem específica, ocasionando alguns tipos de permissões e acessos aos atacantes. Os tipos de ataques estão catalogados pelo nível de violação e, segundo Bace e Mell (2001, p.40), são:

- Confiabilidade: Causam violação da confiabilidade permitindo o atacante ter acesso a dados sem necessidade de autorização.
- Integridade: Causam violação da integridade permitindo o atacante mudar estados do sistema ou qualquer dado residente ou que está passando no sistema.
- Disponibilidade: Causam violação de disponibilidade permitindo o acesso aos recursos do sistema violado quando, onde e da maneira que o atacante necessite.
- Controle: Causam violação de controle que garante ao atacante privilégio sobre a política de controle de acesso do sistema (sem autorização). Estes privilégios podem permitir que o atacante force os demais níveis de violação.

[27] *Cracker* – Pessoa com conhecimentos profundos em sistemas que usa estes conhecimentos com finalidade negativa como crimes eletrônicos.

3.3.1. Engenharia Social

A engenharia humana é a forma de obtenção de dados mais perigosa e difícil de ser combatida, porque é feita através de contato direto com pessoas para coleta de informações importantes.

O atacante pode reunir informações sigilosas sobre usuários – *username* e senha de acesso – apenas utilizando ligações para pessoas-chave dentro de setores de uma instituição e até mesmo coleta-las através do setor de informática da empresa. Dessa forma, a política de segurança de uma instituição tem que ser bem explícita e clara sobre casos desses tipos.

Outro ponto importante na obtenção de informações utilizando engenharia humana está ligada a ex-funcionários. Pessoas com má intenção podem facilmente conseguir senhas para acesso apenas ligando para ex-funcionários; inclusive aqueles que não têm seus passes desabilitados ou apagados podem utilizá-los com más intenções.

3.3.2. Exploração de Erros e Vulnerabilidades

A exploração de erros e vulnerabilidades de sistemas é um dos meios mais comuns utilizados para início de ataques. O monitoramento e constante pesquisa sobre vulnerabilidades dos sistemas instalados são as melhores maneiras de garantir a segurança em altos níveis. Fazem parte destes procedimentos a atualização de versões dos programas, a instalação de correções de segurança e o uso de sistemas de notícias sobre segurança.

Segundo Campello e Webers (em fase de elaboração), os principais tipos de vulnerabilidades citados no relatório elaborado pelo NIST (*National Institute of Standards and Technology*), são os seguintes:

- ❖ Erro na validação de entrada: Erros causados por entradas de dados indevidamente tratadas, ou seja, conjuntos de dados não especificados que geram resultados inesperados quando inseridos no sistema. O mais comum é denominado de *buffer overflow*: os dados recebidos pelo sistema são maiores que o esperado; sem verificação por parte do sistema, esses dados extrapolam espaços reservados à alocação, invadindo posições de memória

reservados a outros tipos de dados, podendo causar execução de rotinas não autorizadas ou paralisações.

- ❖ Erro na validação de acesso: Erros de projeto ou implementação que ocasionam em falhas nos mecanismos de controle de acesso.
- ❖ Erro de tratamento de exceções: Erros ocasionados por má manipulação de exceções ocorridas.
- ❖ Erro de ambiente: Problemas de segurança em programas causados pelo ambiente onde ele está inserido.
- ❖ Erro de configuração: Vulnerabilidades ocasionadas por má configuração dos sistemas. Os sistemas tornam-se inseguros por erro na configuração efetuada pelo usuário.
- ❖ Condições de corrida: Ocorrem quando existe um atraso entre o momento em que o sistema verifica se uma operação é permitida e a efetivação dessa operação. Nesse espaço de tempo, ações ilegais podem ser executadas.

A vulnerabilidade de serviços geralmente é explorada pelos atacantes através da união com outros tipos de ataques. “Fingindo” estar em outra máquina (falsificação de endereços IP, item 5), o atacante pode facilmente utilizar os serviços da máquina vítima utilizando os serviços de *rlogin*, *rsh*, etc., pois tais serviços possuem arquivos de relações de confiança em máquinas (*/etc/hosts.equiv* e *~/.rhosts*), excluindo a necessidade de senhas para validação por parte de usuários autenticados em alguma destas máquinas confiáveis. Outro programa bastante inseguro é o *finger*. Usuários autenticados num Sistema Operacional, podem facilmente utilizar o programa *finger* e conseguir informações importantes sobre usuários conectados no sistema naquele momento ou conseguir informações mais detalhadas destes (nome, *username*, etc.).

3.3.3. Bisbilhotagem de Pacotes (*Packet Sniffing*)

A checagem de pacotes é uma técnica utilizada em casos benignos e malignos. Todos os pacotes IP que trafegam, devido à forma pela qual as redes foram projetadas, passam por todas as máquinas, com raras exceções – como nos casos de redes que utilizam *switches*. Diante desse tipo de configuração, uma máquina qualquer pode conferir todos os pacotes, mesmo aqueles que não se destinam a sua máquina. Este

processo de “farejamento” (*sniffing*) da rede é chamado de bisbilhotagem de pacotes, do inglês *packet sniffing*. Existem dois tipos de bisbilhotagem e em ambos os casos, é necessário que o atacante já tenha obtido acesso em alguma máquina na rede interna. Os tipos de bisbilhotagem de pacotes são: [1] Passivo, onde só são observados os pacotes sem alteração; e [2] Ativo, onde o atacante pode alterar e gerar novos pacotes.

3.3.4. Varredores de Portas (*Port Scanners*)

A varredura de portas TCP e UDP é o primeiro passo tomado quando existe a intenção de ataque a uma máquina sem que se saiba quais são os serviços instalados. Com a única intenção de descobrir os serviços que estão funcionando e, em alguns casos, qual o Sistema Operacional da máquina vítima, a varredura de portas não é considerada um ataque, porém é vista como um indício do início de um.

As formas para varredura de portas evoluíram bastante ao longo dos últimos anos, porém sempre estarão baseadas no seu princípio básico: abrir conexões em uma faixa de portas definidas pelo atacante na máquina ou rede de máquinas vítima. Este processo pode ser observado na Figura 24.

```
11:56:20.442740 connect.scanner.net.1141 > victim.cablemodem.com.21:
S 929641:929641(0) win 8192 <mss 536,nop,nop,sackOK> (DF)

11:56:21.191786 victim.cablemodem.com.21 > connect.scanner.net.1141:
S 779881634:779881634(0) ack 929642 win 8576 <mss 1460> (DF)

11:56:21.201490 connect.scanner.net.1141 > victim.cablemodem.com.21:
. ack 1 win 8576 (DF)

11:56:23.954930 connect.scanner.net.1144 > victim.cablemodem.com.37:
S 932103:932103(0) win 8192 <mss 536,nop,nop,sackOK> (DF)

11:56:24.647238 victim.cablemodem.com.37 > connect.scanner.net.1144:
R 0:0(0) ack 1 win 0
```

Figura 25 – Exemplo de varredura de portas.

Na Figura 24, o atacante *connect.scanner.net* faz uma conexão completa (aperto de mão de três vias) com a vítima na porta 21 e recebe uma mensagem dela que não existem serviços esperando conexões na porta 37. Este é a maneira mais antiga para varredura de portas.

Existem outras formas mais modernas para varredura de portas, uma delas é denominada *TCP SYN* ou “*half connect*”, onde o atacante reinicia a conexão logo após o recebimento do pacote de confirmação de conexão da vítima. Outro meio de varredura de portas é o *FIN scan*, na qual o atacante envia pacotes de finalização de conexão sem ter tido nenhum tipo de conexão prévia entre ambos.

Quando a rede está protegida por filtros de pacotes (Capítulo 3, Seção 1.1), uma forma eficiente para que a varredura de pacotes não seja barrada pelo *firewall* (*screening router*) é utilizar a varredura de portas invisível (*stealth scan*). Esse tipo de varredura consiste no envio de pacotes como se a conexão já estivesse sido efetuada (os filtros geralmente negam pacotes de abertura de conexão). Conforme a resposta da vítima, o atacante saberá se existe ou não serviço ativo na porta.

3.3.5. Falsificação de endereço IP (*Source Address Spoofing*)

A falsificação de endereços IP é utilizada por atacantes para forjarem a sua existência, dificultar o rastreamento pelos administradores de rede, ataques de negação de serviço, uso de privilégios de outras máquinas, etc. Consiste em enviar pacotes de com endereço de origem forjado para alguma máquina ou conjunto de máquinas.

O uso mais famoso de falsificação de IP foi utilizado por Kevin Mitnick (1995) para “seqüestrar” (item 7) a conexão do também hacker Tsutomu Shimomura. Mitnick, forjando estar em uma máquina de Shimomura (forjando o seu endereço IP), invadiu uma máquina na rede de confiança de Shimomura utilizando os serviços de *rlogin* da máquina vítima.

A falsificação é utilizada como ferramenta em vários tipos de ataques, possibilitando que sejam mais eficientes, pois enganam, na maioria dos casos, roteadores e *firewalls*, fingindo ser quem não é.

Para criação de um octeto IP com endereço de origem falso é necessário apenas que seja alterado o endereço de origem e então o octeto é enviado à rede pelo dispositivo de rede.

3.3.6. Negação de Serviço (*Denial of Service*)

Os ataques de Negação de Serviços são utilizados para paralisação temporária dos serviços de máquinas numa rede. O ataque é baseado no excesso de tráfego para um computador (o bastante para que não consiga responder) ou esgotando o processamento de um servidor por outros meios.

O primeiro ataque DoS[28] registrado foi em 1988, quando Robert Morris Jr., na época defendendo sua tese de doutorado na Universidade de Cornell, lançou um programa que se reproduzia através de vulnerabilidades dos serviços de e-mail (*sendmail*), *rsh* e *fingerd* dos computadores ligados à internet. O programa atuava como um "verme" (do inglês, *worm*), rastejando entres os servidores infectados e procurando novos computadores para infectar. O problema que Morris não havia previsto é que o *worm* não era capaz de se detectar na própria máquina, exaurindo os recursos computacionais das máquinas infectadas pela re-infecção contínua.

Um dos ataques DoS mais conhecidos é o *SYN Flood*, onde a máquina vítima recebe uma quantidade enorme de requisições de conexão (SYN), onde o endereço de origem é forjado (*flood*), estourando o *buffer* de conexões e retirando a máquina temporariamente de funcionamento. Esse ataque utiliza a fragilidade do aperto de mão de três vias do TCP (Capítulo 2, Seção 3), enviando à vítima apenas o primeiro pacote do processo de conexão e ainda com endereço IP forjado (muitas vezes inexistente), fazendo com que seja perdido tempo na espera de resposta da máquina forjada e inexistente (o servidor envia o pacote de confirmação de conexão, segundo passo do aperto de mão de três vias), abarrotando o *buffer* de conexões da vítima.

A evolução do DoS é o *Distributed Denial of Service* (DDoS), do inglês, Negação de Serviço Distribuído. O DDoS consiste em utilizar mais de uma máquina para causar a negação dos serviços das máquinas vítima. Existem várias maneiras sofisticadas para a interrupção dos serviços e uma delas é o ataque denominado de *Smurf*, no qual são enviados pacotes ICMP de difusão numa rede com o endereço de origem forjado (com o endereço da vítima). Ao receberem a mensagem, as máquinas respondem à vítima, interrompendo seus serviços com mensagens ICMP de resposta.

Outro tipo de ataque DDoS largamente utilizado para a paralisação e congestionamento de redes utiliza agentes para envio da mensagem. Conforme Solha,

---

[28] Abreviatura de *Denial of Service* – Negação de Serviço em inglês.

Cicilini e Piccolini (2000), o ataque DDoS tem quatro personagens: [1] atacante, que coordena o ataque; [2] *master*, máquina que recebe os parâmetros de ataque e repassa aos agentes; [3] agente, máquina que realmente concretiza o ataque e [4] a vítima. Ainda sobre o ataque, são comentados os três passos sobre o ataque: [1] intrusão em massa, onde o atacante coleta e invade máquinas através das mais variadas formas de invasão, elegendo o *master* e os agentes; [2] instalação do software DDoS no *master* (cliente) e nos agentes (*daemon*) e [3] o ataque propriamente dito que é coordenado pelo atacante, enviado ao *master* e repassado aos agentes que atacam a vítima. A Figura 25 exemplifica a estrutura do ataque DDoS.
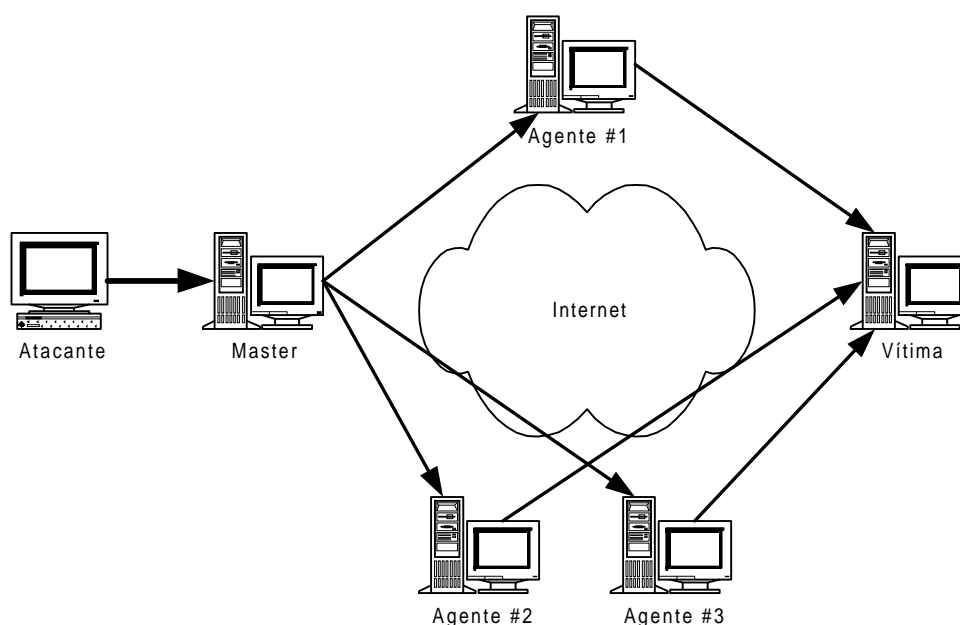
Figura 26 – Ataque DDoS.

Existem inúmeras ferramentas DDoS cadastradas no CERT, todas muito sofisticadas e cada vez mais amigáveis com o usuário. As principais ferramentas cadastradas são: *Trin00*, *Trible Flood Network (TFN)*, *Trible Floodnet 2k (TFN2K)*, *Stacheldraht*, *dscan*, *Blitznet*, *Fapi*, *Trank*, *Shaft.*

3.3.7. Ataques de Seqüestro de Conexões (*Hijacking Attacks*)

Os ataques de seqüestro de conexão entre máquinas são os ataques mais complexos dentre os demais e exploram bastante as vulnerabilidades dos protocolos de rede. Neste tipo de ataque, existe uma terceira máquina, entre duas máquinas que estão trocando

informações, que irá "seqüestrar" a comunicação. Esta máquina é denominada de "homem no meio", do inglês, *man-in-a-middle*.

No ataque de seqüestro de conexão, a máquina seqüestradora irá quebrar a seqüência correta de conexão entre duas máquinas. Para isso é necessário que a máquina atacante utilize técnicas para prever ou bisbilhotar os números de seqüência que são trocados conjuntamente com as informações entre as duas máquinas vítima (os números de seqüência numa conexão TCP). Já com a seqüência da conexão das duas máquinas, a máquina atacante retira uma delas do ar com ataques por negação de serviço (item 6) e passa a enviar pacotes, como se fosse a máquina que foi interrompida, por meio de falsificação de endereço IP de origem. Está seqüestrada a conexão. A máquina atacante pode ter acesso às informações confidenciais que só iriam ser trocadas entre as duas máquinas vítima.

3.3.8. Vulnerabilidades dos Protocolos de Rede

O protocolo de rede IP e os protocolos a ele relacionados foram projetados sem as devidas atenções sobre segurança, criando alguns problemas neste âmbito. Apesar de eficientes e de certa forma simples, os protocolos de rede internet são também suscetíveis a falhas perigosas.

- As falhas por falta de checagem de cabeçalhos eram, e ainda são em alguns casos, pontos importantes. Ataques como o *LAND*, onde os endereços de origem e de destino são iguais, paralisavam diversos tipos de Sistemas Operacionais (e ainda paralisam).
- O *Ping of Death*, que consistem em enviar pacotes com tamanho maiores de 65535 octetos (geralmente pacotes ICMP, daí o nome *ping*) também foram, por um bom tempo, problemas na internet. O tamanho máximo permitido de um pacote no protocolo IP é de 65353 octetos.
- O *Teardrop*, ataque que explora uma falha de implementação na montagem de fragmentos de um pacote IP. Consiste em colocar no endereço de início de um fragmento um valor posterior ao endereço de fim do fragmento. Este ataque faz o Sistema Operacional se desestabilizar apenas tentando remontar os fragmentos.

Estes foram alguns exemplos de como o protocolo pode ser usado de forma destrutiva num ambiente de rede, re-afirmando sua fragilidade de projeto.

Outros protocolos bastante frágeis são os protocolos para redes de Sistemas Operacionais *Windows*. Como são protocolos para redes internas é necessário o cuidado para que nenhuma máquina ou pessoa tenha acesso a serviços baseados nestes protocolos via rede externa. Geralmente encapsulados no protocolo IP, estes protocolos funcionam como serviços nas portas de comunicação 137, 138 e 139 (Anexo B).

3.3.9. Falsos Ataques e Alertas

Os falsos ataques e alertas são suspeitas levantadas por meio de Sistemas de Detecção de Intrusos, que não passam de simples suspeitas.

Como não é possível acertar todos os diagnósticos, os IDS podem gerar um falso alerta sobre alguns movimentos permitidos. Um exemplo bem interessante disso é o *coordinated traceroute*, que é um mecanismo utilizado por alguns provedores de serviços para cálculo do menor caminho de resposta aos seus clientes.

Outro tipo de falso ataque é denominado *hoax* (do inglês, brincadeira, peça, etc.) que tem como único propósito criar pânico a quem administra a máquina ou rede alvo. Este tipo de brincadeira está se tornando comum e já está sendo considerado como forma de terrorismo por autoridades mundiais.

4. Estudo de Caso

O estudo sobre a defesa de redes de computadores comprovou que as soluções de defesa são moldadas às situações, às condições financeiras e à importância das instituições. Não seria possível a criação de uma corrente de regras padrão para certificar que a rede de computadores está segura; a segurança advém do uso do bom senso e cuidados padrão. A padronização sobre segurança está nos métodos de proteção, que incluem as ferramentas e procedimentos que devem ser adotados, não podendo ser aplicadas no modo da implantação.

A defesa de rede é privada e construída diante das circunstâncias de necessidades; portanto, o estudo de códigos para aplicação destas regras depende de um contexto que criem as condições para essa aplicação.

O exemplo do uso de códigos para defesa de redes de computadores deste trabalho será baseado num estudo de um caso fictício de uma empresa registrada como Alfa.

4.1. Visão Geral

O estudo de caso apresenta o ambiente de rede da empresa Alfa.

A Alfa é uma empresa de Serviços Internet (*Internet Services Provider*) e de Serviços de Atendimento ao Público (*Call Center*). A Alfa tem contrato de prestação de serviços com a empresa Beta que consiste no aluguel de espaço físico para colocação de uma máquina provedora de serviços de páginas HTML e dinâmicas e contrato de prestação de serviços com a empresa Gama, que consiste em 20 posições de atendimento a clientes num *Call Center* (de propriedade da Alfa) com acesso ao banco de dados corporativo localizado na sede de Gama.

A Alfa tem um escritório administrativo no mesmo prédio das instalações dos servidores Internet e do seu *Call Center*. Existe um servidor interno para banco de dados das informações corporativas da empresa e servidores de produção da equipe de desenvolvimento e criação. O servidor de banco de dados corporativo da Alfa é acessado pelos seus servidores Internet e também pelos seus servidores internos.

A política de segurança para seus usuários no escritório da sede da Alfa e para os seus funcionários do *Call Center* é a seguinte (simplificada):

1 – O acesso à internet pelos funcionários é restrito: é permitindo somente navegação em páginas internet.

2 – É permitido somente o uso de serviço de correio eletrônico da instituição.

3 – As estações de trabalho do *Call Center* não têm acesso aos serviços internet.

4 – É proibido o uso de qualquer serviço que não esteja declarado acima, com exceção dos serviços autorizados pela diretoria da empresa.

Baseados nos serviços prestados por Alfa e na política de segurança, a estrutura de proteção é composta por um *firewall* com arquitetura *Screened Subnet* (Capítulo 3, Seção 1.4) ligeiramente alterada (dividindo a rede periférica em duas redes – *backbone* interno e rede desmilitarizada) e por um Sistema de Detecção de Intrusos – IDS (Capítulo 3, Seção 2), que faz a análise da rede desmilitarizada.
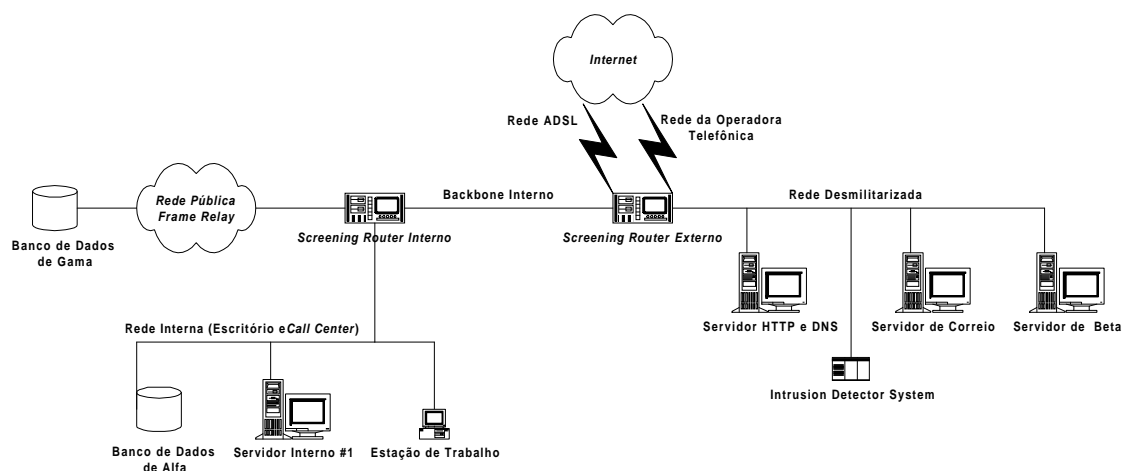
Figura 27 – Rede da Alfa.

As redes periférica e interna juntamente com os *Screening Routers* constituem a arquitetura de *firewall Screened Subnet*. A rede *Frame Relay*[29] está ligada ao roteador interno e serve de ligação ao servidor de banco de dados do Gama.

O acesso da rede interna à internet está restrito por meio de filtro de pacotes (Capítulo 3, Seção 1.1) às portas 80 e 443 utilizando a ligação à rede ADSL. O segundo link à internet é somente para servidores da rede desmilitarizada e somente estarão liberadas as portas dos serviços que são providos pelos servidores (filtro de pacotes).

[29] Protocolo de enlace utilizado em redes chaveadas que utilizam múltiplos circuitos virtuais.

Qualquer outro tipo de tráfego da rede interna ou da rede *Frame Relay* ao roteador externo é negado.

Para filtro de pacotes nos *Screening Routers* é utilizado a ferramenta *IP Tables* (Anexo C) e para detecção de intrusos a ferramenta *Snort* (Anexo D). Os códigos das regras de filtragem e para detecção de intrusos estão no anexo E.

5. Conclusão

A consciência da necessidade de proteção dos dados é um fator crescente entre as instituições. O Projeto de Segurança Eletrônica deixa de ser um gasto adicional e torna-se um dos maiores investimentos para instituições, consolidando a importância deste tópico na construção de empresas que utilizam serviços digitais.

Esta preocupação é o reflexo nos serviços eletrônicos, digitais e informatizados do crescimento de investimentos em segurança no mundo, gerando o aparecimento de normas para implantação da segurança eletrônica em diversos países. Atualmente, há uma infinidade de regulamentações que determinam os procedimentos para a implantação da segurança da tecnologia da informação. Dentre elas destacam-se as pioneiras normas do comitê britânico *Commercial Computer Security Centre* (CCSC) denominadas BS 7799:1995 e as suas revisões nos anos de 1998, 1999 e 2000 e as normas técnicas baseadas na norma da comunidade britânica e editadas no Brasil pela Associação Brasileira de Normas Técnicas (ABNT) determinadas pelo comitê ISO (*International Standardization Organization*) da série ISO 17799:2000.

A maioria dos procedimentos adotados nas diversas normas técnicas mundiais tem como base o estudo das principais ameaças e vulnerabilidades dos sistemas de comunicação eletrônica. Ao longo destes estudos, foi constatado que a modificação ou criação de simples procedimentos de segurança aumentam consideravelmente as defesas dos sistemas das redes institucionais. Estes procedimentos vão desde a discussão de medidas simples do uso de ferramentas dentro das instituições (políticas de segurança institucional e manual do usuário) até a instalação de mecanismos de obstrução da entrada de dados não autorizados ou desconhecidos (*firewalls*), com a utilização de sistemas para monitoração destas medidas (Sistemas de Detecção de Intrusos).

A implantação e administração destes procedimentos, no entanto, são trabalhosos e requerem conhecimento e tempo, sendo suscetível a falhas humanas, uma vez que o administrador faz a verificação e análise de uma quantidade grande de informações sobre possíveis invasões, ataques e alarmes. A complexidade da análise aliada à falta de experiência do administrador, portanto, pode ter conseqüências catastróficas, gerando uma necessidade eminente de criação de ferramentas para auxílio de avaliação dos arquivos de registros – que são bastante simples para garantir a

performance dos sistemas – e de ferramentas que também facilitem a criação de novas regras de proteção – também dependentes da intervenção humana.

Em relação a criação e manutenção de Sistemas de defesa, os maiores problemas na implantação da segurança são as falhas nos projetos dos protocolos de comunicação, gerando uma quantidade significativa de brechas para a transposição dos modelos normais de uso e conseqüentemente da segurança dos sistemas que estão baseados nestes protocolos. Os protocolos de rede neste trabalho mostrados – tais como o IP, TCP, UDP e o ICMP – são os mais utilizados na rede mundial de computadores (Internet) e em todos eles há problemas que implicam na transposição da segurança. Como a interligação das instituições é feita através destes protocolos e utilizando a Internet, o perigo é constante. Os fatores que agravam ainda mais este quadro são os serviços de rede que também apresentam falhas de projeto e estão em níveis mais altos, facilitando a manipulação dos dados por pessoas mal intencionadas.

É também evidente a tendência das defesas de computadores e redes por meio da Inteligência Artificial. Os *firewalls* e Sistemas de Detecção de Intrusos modernos utilizam um dos fundamentos da Inteligência Artificial, que é o reconhecimento de padrões. No caso da defesa da informação, os padrões a serem reconhecidos (assinaturas de ataque) são as diversas formas de tentativas de transposição da segurança, bisbilhotagem alheia dos sistemas internos, tentativas de interrupção dos serviços e utilização de componentes nocivos para a rede, como Vírus, Cavalos de Tróia, *Back-doors*, etc.

Os Sistemas de Detecção de Intrusos mais elaborados também já implantam uma forma rudimentar de detecção de invasões baseadas em redes neurais. A maneira na qual são conduzidos os estudos parece ser promissora e tende a chegar em níveis nos quais são criadas formas de defesa utilizando agentes móveis e inteligentes (também da Inteligência Artificial) capazes de detectar uma falha eletrônica sem a intervenção humana e podendo agir sozinhos, aprendendo como se defender e destruir a causa dos problemas, como Sistemas Imunológicos tradicionais.

Por último, é imprescindível observar que a utilização das mais diversas normas e métodos para defesas das informações não são eternas e somente serão seguras até quando forem violadas. É necessário monitorar constantemente as soluções de defesas definidas para que surpresas desagradáveis não aconteçam, desestabilizando os sistemas

por completo. A criação de planos de emergência também são necessários, pois prevêem casos de invasões, criam procedimentos para tais circunstâncias, bem como diversos tipos de testes, desenvolvendo situações onde o atacante tem pouco conhecimento dos sistemas e ambiente (simulando um atacante externo) e situações onde o atacante tem conhecimentos profundos sobre os sistemas e estruturas dos sistemas (simulando um atacante interno).

A finalidade deste trabalho foi apresentar um estudo de caso sobre segurança de redes destacando a importância do uso de regras e procedimentos na detecção de e prevenção de ameaças eletrônicas.

6. Referências

BERNSTEIN, Terry, BLUMANI, Anish B., SCHULTZ, Eugene, SIEGEL, Carol A. ***Internet Security for Business***. Nova Iorque: Wiley Computer Publishing, 1996.

BLACK, Uyless. ***TCP/IP and Related Protocols***. 2 ed. Nova Iorque: McGraw-Hill, 1994.

SIYAN, Karanjit, HARE, Christopher. ***Internet Firewalls and Network Security***. Indianapolis: New Riders Publishing, 1995.

GARFINKEL, Simson, SPAFFORD, Gene. ***Practical UNIX & Internet Security***. 2 ed. O'Reilly & Associates, 1996.

HUNT, Craig. ***TCP/IP Network Administration***. 2 ed. O'Reilly & Associates, 1997.

NAUGLE, Matthew G. ***Illustrated TCP/IP***. Wiley Computer Publishing, 1998.

CHAPMAN, D. Brent, ZWICKY, Elizabeth D. ***Building Internet Firewalls***. 1 ed. O'Reilly & Associates, 1995.

____ 2 ed. O'Reilly & Associates, 2000.

NATIONAL SECURITY AGENCY (USA). Systems and Network Attack Center. ***The 60 Minutes Network Security Guide (First Steps Towards a Secure Network Environment).*** Version 1.0. Ft. Meade. 2001.

BEJTLICH, Richard. ***Interpreting Network Traffic: A Network Intrusion Detector's Look at Suspicious Events.*** Version 2.8. http://home.satx.rr.com/bejtlich. 2000.

GREEN, John, MARCHETTE, David, NORTHCUTT, Stephen. ***Analysis Techniques for Detecting Coordinated Attacks and Probes.*** 2000.

SOLHA, Liliana Esther Velásquez Alegre, CICILINI, Renata, PICCOLINI, Jacomo Dimmit Boca. Centro de Atendimento a Incidentes de Segurança da RNP. ***Tudo que Você Precisa Saber Sobre os Ataques DDoS.*** NewsGeneration. Volume 4, Número 2. http://www.rnp.br/newsgen. 2000.

GRAHAM, Robert. ***FAQ: Network Intrusion Detection Systems.*** Version 0.8.3. http://www.robertgraham.com/pubs/network-intrusion-detection.html. 2000.

ROESCH, Martin. ***Snort Users Manual***. Version 1.8.2. http://www.snort.org. 2001.

__. ***Snort - Lightweight Intrusion Detection for Networks***. http://www.snort.org. 2001.

RUSSEL, Rusty. ***Linux 2.4 NAT HOWTO***. Revision 1.15. http://netfilter.samba.org. 2001.

__. ***Linux Netfilter Hacking HOWTO***. Revision 1.10. http://netfilter.samba.org. 2001.

__. ***Packet Filtering HOWTO***. Revision 1.21. http://netfilter.samba.org. 2001.

JUNIOR, Rafael T. De Sousa, PUTTINI, Ricardo S. ***Criptografia e Segurança de Redes de Computadores***. http://www.redes.unb.br/security/firewall/firewall.html.

7. Anexos

Anexo A. Tipos de Requisições ICMP e suas Opções.

Kurt Seifried, kurt@seifried.org

September 25, 2001 - 53 entries

http://www.seifried.org/security/ports/

ICMP Code	***ICMP Code Name***	
0	*echo*	
Code Type	***Type Name***	***Type Description***
0	*echo-reply*	*RFC 792 - for echo reply message*
ICMP Code	***ICMP Code Name***	
3	*unreachable*	
Code Type	***Type Name***	***Type Description***
0	*net-unreachable*	*RFC 792 - net unreachable*
1	*host-unreachable*	*RFC 792 - host unreachable*
2	*protocol-unreachable*	*RFC 792 - protocol unreachable*
3	*port-unreachable*	*RFC 792 - port unreachable*
4	*fragmentation-df-set*	*RFC 792 - fragmentation needed and DF set*
5	*source-route-failed*	*RFC 792 - source route failed*
6	*dest-network-unknown*	*RFC 792 - Destination Network Unknown*
7	*dest-port-unknown*	*RFC 792 - Destination Host Unknown*
8	*source-host-isolated*	*RFC 792 - Source Host Isolated*
9	*network-admin*	*RFC 792 - Communication with Destination Network is Administratively Prohibited Communication with Destination Network is Administratively Prohibited*
10	*host-admin*	*RFC 792 - Communication with Destination Host is Administratively Prohibited*
11	*network-service*	*RFC 792 - Destination Network Unreachable for Type of Service*
12	*host-service*	*RFC 792 - Destination Host Unreachable for Type of Service*
13	*com-admin-prohibited*	*RFC 1812 - Communication Administratively Prohibited - generated if a router cannot forward a packet due to administrative filtering*
14	*host-precedence-violation*	*RFC 1812 - Host Precedence Violation. Sent by the first hop router to a host to indicate that a requested precedence is not permitted for the articular combination of source/destination host or network, upper layer protocol, and source/destination port*
15	*precedence-cuttof-in-effect*	*RFC 1812 - Precedence cutoff in effect. The network operators have imposed a minimum level of precedence required for operation, the datagram was sent with a precedence below this level*
ICMP Code	***ICMP Code Name***	
4	*quench*	

Code Type	***Type Name***	***Type Description***
0	*source-quench*	*RFC 792 - source quench (slow down!)*
ICMP Code	***ICMP Code Name***	
5	*redirect*	
Code Type	***Type Name***	***Type Description***
0	*redirect-network*	*RFC 792 - Redirect datagrams for the Network (or subnet)*
1	*redirect-host*	*RFC 792 - Redirect datagrams for the Host*
2	*redirect-service-network*	*RFC 792 - Redirect datagrams for the Type of Service and Network*
3	*redirect-service-host*	*RFC 792 - Redirect datagrams for the Type of Service and Host*
ICMP Code	***ICMP Code Name***	
6	*alternate*	
Code Type	***Type Name***	***Type Description***
0	*alternate-host-address*	*JBP - Alternate address for host*
ICMP Code	***ICMP Code Name***	
8	*echo*	
Code Type	***Type Name***	***Type Description***
0	*echo-request*	*RFC 792 - for echo message*
ICMP Code	***ICMP Code Name***	
9	*router*	
Code Type	***Type Name***	***Type Description***
0	*router-advertisement*	*RFC 1256 - Router advertisement*
ICMP Code	***ICMP Code Name***	
10	*router*	
Code Type	***Type Name***	***Type Description***
0	*router-selection*	*RFC 1256 - Router selection*
ICMP Code	***ICMP Code Name***	
11	*exceeded*	
Code Type	***Type Name***	***Type Description***
0	*ttl-exceeded*	*RFC 792 - time to live exceeded in transit*
1	*fragment-reassembly-exceeded*	*RFC 792 - fragment reassembly time exceeded*
ICMP Code	***ICMP Code Name***	
12	*error*	
Code Type	***Type Name***	***Type Description***
0	*pointer-error*	*RFC 792 - pointer indicates the error*
1	*missing-option*	*RFC 792 - Missing a Required Option*
2	*bad-length*	*RFC 792 - Bad Length*
ICMP Code	***ICMP Code Name***	
13	*timestamp*	
Code Type	***Type Name***	***Type Description***
0	*timestamp-request*	*RFC 792 - for timestamp message*
ICMP Code	***ICMP Code Name***	
14	*timestamp*	
Code Type	***Type Name***	***Type Description***
0	*timestamp-reply*	*RFC 792 - for timestamp reply message*
ICMP Code	***ICMP Code Name***	

15	*information*	
Code Type	***Type Name***	***Type Description***
0	*info-request*	*RFC 792 - for information request message*
ICMP Code	***ICMP Code Name***	
16	*information*	
Code Type	***Type Name***	***Type Description***
0	*info-reply*	*RFC 792 - for information reply message*
ICMP Code	***ICMP Code Name***	
17	*mask*	
Code Type	***Type Name***	***Type Description***
0	*mask-request*	*RFC 950 - Address Mask Request*
ICMP Code	***ICMP Code Name***	
18	*mask*	
Code Type	***Type Name***	***Type Description***
0	*mask-reply*	*RFC 950 - Address Mask Reply*
ICMP Code	***ICMP Code Name***	
30	*traceroute*	
Code Type	***Type Name***	***Type Description***
0	*traceroute-forwarded*	*RFC 1393 - Traceroute - Outbound Packet successfully forwarded*
1	*packet-discarded*	*RFC 1393 - traceroute - No route for Outbound Packet; packet discarded*
ICMP Code	***ICMP Code Name***	
31	*datagram*	
Code Type	***Type Name***	***Type Description***
0	*datagram-conversion-error*	*RFC 1475 - Datagram Conversion Error*
ICMP Code	***ICMP Code Name***	
32	*mobile*	
Code Type	***Type Name***	***Type Description***
0	*mobile-host-redirect*	*David Johnson - Mobile Host Redirect*
ICMP Code	***ICMP Code Name***	
33	*ipv6-request*	
Code Type	***Type Name***	***Type Description***
0	*ipv6-where-are-you*	*Bill Simpson -IPv6 Where-Are-You*
ICMP Code	***ICMP Code Name***	
34	*ipv6-reply*	
Code Type	***Type Name***	***Type Description***
0	*ipv6-here-I-am*	*Bill Simpson - IPv6 I-Am-Here*
ICMP Code	***ICMP Code Name***	
35	*mobile*	
Code Type	***Type Name***	***Type Description***
0	*mobile-registration-request*	*Bill Simpson - Mobile Registration Request*
ICMP Code	***ICMP Code Name***	
36	*mobile*	
Code Type	***Type Name***	***Type Description***
0	*mobile-registration-reply*	*Bill Simpson - Mobile Registration Reply*
ICMP Code	***ICMP Code Name***	
37	*domain-name*	

Code Type	***Type Name***	***Type Description***
0	*domain-name-request*	*RFC 1788 - icmp domain name request*

ICMP Code	***ICMP Code Name***
38	*domain-name*

Code Type	***Type Name***	***Type Description***
0	*domain-name-reply*	*RFC 1788 - icmp domain name reply*

ICMP Code	***ICMP Code Name***
40	*security*

Code Type	***Type Name***	***Type Description***
0	*bad-spi*	*RFC 2521 - Bad SPI*
1	*authentication-failed*	*RFC 2521 - Authentication Failed*
2	*decompression-failed*	*RFC 2521 - Decompression Failed*
3	*decryption-failed*	*RFC 2521 - Decryption Failed*
4	*need-authentication*	*RFC 2521 - Need Authentication*
5	*need-authorization*	*RFC 2521 - Need Authorization*

Anexo B. Serviços de Rede, TCP e UDP (Resumo com os Principais Serviços).

Kurt Seifried, kurt@seifried.org

September 23, 2001 - 6983 entries

http://www.seifried.org/security/ports/

Service-name	Port/Protocol	Comment
daytime	13/tcp	Daytime (RFC 867)
daytime	13/udp	Daytime (RFC 867)
netstat	15/tcp	Network status
ftpdata	20/tcp	File Transfer Protocol [Default Data]
ftpdata	20/udp	File Transfer Protocol [Default Data]
ftp	21/udp	File Transfer Protocol [Control]
ftp	21/tcp	File Transfer Protocol [Control]
ssh	22/tcp	SSH Remote Login Protocol
ssh	22/udp	SSH Remote Login Protocol | pcanywhere
telnet	23/tcp	Telnet
telnet	23/udp	Telnet
smtp	25/tcp	Simple Mail Transfer Protocol
smtp	25/udp	Simple Mail Transfer Protocol
time	37/tcp	timserver Time
time	37/udp	timserver Time
whois	43/tcp	whois Who Is
whois	43/udp	whois Who Is
domain	53/tcp	Domain Name Server
domain	53/udp	Domain Name Server
dhcp-bootps	67/udp	DHCP|Bootstrap Protocol Server
dhcp-bootps	67/tcp	DHCP|Bootstrap Protocol Server
bootpc	68/tcp	Bootstrap Protocol Client
bootpc	68/udp	Bootstrap Protocol Client
tftp	69/tcp	Trivial File Transfer
tftp	69/udp	Trivial File Transfer
gopher	70/tcp	Gopher
gopher	70/udp	Gopher
finger	79/tcp	Finger
finger	79/udp	Finger
www-http	80/tcp	World Wide Web HTTP
www-http	80/udp	World Wide Web HTTP
rtelnet	107/udp	Remote Telnet Service

rtelnet	107/tcp	Remote Telnet Service
pop-3	110/udp	postoffice Post Office Protocol - Version 3
pop-3	110/tcp	postoffice Post Office Protocol - Version 3
rpc	111/tcp	SUN Remote Procedure Call
rpc	111/udp	SUN Remote Procedure Call
sqlserv	118/tcp	SQL Services
sqlserv	118/udp	SQL Services
nntp	119/tcp	Network News Transfer Protocol Usenet
nntp	119/udp	Network News Transfer Protocol Usenet
ntp	123/udp	Network Time Protocol
ntp	123/tcp	Network Time Protocol
password	129/udp	Password Generator Protocol
password	129/tcp	Password Generator Protocol
netbios-ns	137/udp	NETBIOS Name Service nbns
netbios-ns	137/tcp	NETBIOS Name Service nbns
netbios-dgm	138/udp	NETBIOS Datagram Service
netbios-dgm	138/tcp	NETBIOS Datagram Service
netbios-ssn	139/tcp	NETBIOS session service
netbios-ssn	139/udp	NETBIOS Session Service
imap	143/udp	Interim Mail Access Protocol v2 | Internet Message
imap	143/tcp	Interim Mail Access Protocol v2 | Internet Message
sql-net	150/udp	SQL-NET
sql-net	150/tcp	SQL-NET
sqlsrv	156/tcp	SQL Service
sqlsrv	156/udp	SQL Service
snmp	161/udp	Simple Net Mgmt Proto
snmp	161/tcp	Simple Net Mgmt Proto
snmp-trap	162/udp	snmptrap Traps for SNMP
snmp-trap	162/tcp	snmptrap Traps for SNMP
xdmcp	177/udp	X Display Manager Control Protocol
xdmcp	177/tcp	X Display Manager Control Protocol
bgp	179/tcp	Border Gateway Proto.
bgp	179/udp	Border Gateway Protocol
irc	194/tcp	Internet Relay Chat Protocol
irc	194/udp	Internet Relay Chat Protocol
smux	199/udp	SNMP Unix Multiplexer
smux	199/tcp	SNMP Unix Multiplexer

dbase	217/tcp	dBASE Unix
dbase	217/udp	dBASE Unix
ldap	389/udp	LDAP - Lightweight Directory Access Protocol
ldap	389/tcp	LDAP - Lightweight Directory Access Protocol
netware-ip	396/tcp	Novell Netware over IP
netware-ip	396/udp	Novell Netware over IP
appleqtc	458/tcp	apple quick time
appleqtc	458/udp	apple quick time
kpasswd	464/tcp	Kerberos password changing protocol
kpasswd	464/udp	Kerberos password changing protocol
smtps	465/udp	smtp protocol over TLS|SSL (was ssmtp)
smtps	465/tcp	smtp protocol over TLS|SSL (was ssmtp)
isakmp	500/tcp	IPsec Key Management (ISAKMP|Oakley)
isakmp	500/udp	IPsec Key Management (ISAKMP|Oakley)
exec	512/tcp	BSD rexecd(8) - remote process execution
login	513/tcp	BSD rlogind(8)
who	513/udp	maintains data bases showing who's - whod rwhod(8)
syslog	514/udp	BSD syslogd(8)
shell	514/tcp	BSD rshd(8) - remote command shell cmd
printer	515/tcp	spooler BSD lpd(8) - line printer spooler
talk	517/tcp	BSD talkd(8)
talk	517/udp	BSD talkd(8)
ntalk	518/tcp	(talkd)
ntalk	518/udp	SunOS talkd(8)
utime	519/udp	unixtime
utime	519/tcp	unixtime
route	520/udp	local routing process (on site) | router routed
timed	525/udp	timeserver
timed	525/tcp	timeserver
irc-serv	529/udp	IRC-SERV
irc-serv	529/tcp	IRC-SERV
uucp	540/tcp	uucpd BSD uucpd(8) UUCP service
uucp	540/udp	uucpd BSD uucpd(8) UUCP service
uucp-rlogin	541/tcp	uucp-rlogin | rdist daemon

uucp-rlogin	541/udp	uucp-rlogin
klogin	543/tcp	Kerberized `rlogin' (v5)
klogin	543/udp	Kerberized `rlogin' (v5)
kshell	544/udp	krcmd Kerberized rshell (v5)krcmd
kshell	544/tcp	krcmd Kerberized rshell (v5)krcmd
dhcpv6-client	546/tcp	DHCPv6 Client
dhcpv6-client	547/udp	DHCPv6 Server
whoami	565/tcp	whoami
whoami	565/udp	whoami
imap4-ssl	585/tcp	IMAP4+SSL (use 993 instead)
imap4-ssl	585/udp	IMAP4+SSL (use 993 instead)
password-chg	586/tcp	Password Change
password-chg	586/udp	Password Change
ldaps	636/tcp	ldap protocol over TLS|SSL (was sldap)
ldaps	636/udp	ldap protocol over TLS|SSL (was sldap)
kerberos-adm	749/udp	Kerberos 5 kadmin|changepw
kerberos-adm	749/tcp	Kerberos 5 kadmin|changepw
kerberos-iv	750/udp	kdc kerberos4 Kerberos auth (server) udp
kerberos-iv	750/tcp	kdc kerberos Kerberos authentication--tcp
kerberos-master	751/udp	Kerberos admin server tcp
kerberos-master	751/tcp	Kerberos admin server tcp
icq	1027/tcp	ICQ
icq	1029/tcp	ICQ
trojan	1080/tcp	Socks | wingate proxy
socks	1080/udp	Socks
kazaa	1214/tcp	KAZAA
kazaa	1214/udp	KAZAA
lotusnote	1352/udp	Lotus Note
lotusnote	1352/tcp	Lotus Note
ms-sql-s	1433/tcp	Microsoft-SQL-Server
ms-sql-s	1433/udp	Microsoft-SQL-Server
ms-sql-m	1434/udp	Microsoft-SQL-Monitor
ms-sql-m	1434/tcp	Microsoft-SQL-Monitor
sybase-sqlany	1498/tcp	Sybase SQL Any
sybase-sqlany	1498/udp	Sybase SQL Any
utcd	1506/udp	Universal Time daemon (utcd)
utcd	1506/tcp	Universal Time daemon (utcd)
wins	1512/udp	Microsoft's Windows Internet Name Service

wins	1512/tcp	Microsoft's Windows Internet Name Service
orasrv	1525/udp	oracle or Prospero Directory Service nonpriv
orasrv	1525/tcp	oracle or Prospero Directory Service nonpriv
tlisrv	1527/tcp	oracle
tlisrv	1527/udp	oracle
coauthor	1529/tcp	oracle | prmsd gnatsd cygnus bug tracker
coauthor	1529/udp	oracle
rdb-dbs-disp	1571/tcp	Oracle Remote Data Base
rdb-dbs-disp	1571/udp	Oracle Remote Data Base
oraclenames	1575/tcp	oraclenames
oraclenames	1575/udp	oraclenames
oraclenet8cman	1630/udp	Oracle Net8 Cman
oraclenet8cman	1630/tcp	Oracle Net8 Cman
pptp	1723/tcp	Microsoft Point to Point Tunneling Protocol
pptp	1723/udp	Microsoft Point to Point Tunneling Protocol
radius	1812/udp	RADIUS authentication protocol (RFC 2138)
radius	1812/tcp	RADIUS
radacct	1813/udp	RADIUS accounting protocol (RFC 2139)
radius-acct	1813/tcp	RADIUS Accounting
net8-cman	1830/tcp	Oracle Net8 CMan Admin
net8-cman	1830/udp	Oracle Net8 CMan Admin
sybasedbsynch	2439/udp	SybaseDBSynch
sybasedbsynch	2439/tcp	SybaseDBSynch
postgres	5432/tcp	postgres database server
pcanywheredata	5631/tcp	pcANYWHEREdata
pcanywheredata	5631/udp	pcANYWHEREdata
pcanywherestat	5632/udp	pcANYWHEREstat
pcanywhere	5632/tcp	PC Anywhere | pcANYWHEREstat
x11	6000/tcp	X Window System | The Thing Trojan
x11	6000/udp	X Window System
x11	6001/tcp	X Window System
x11	6001/udp	X Window System
x11	6002/tcp	X Window System
x11	6002/udp	X Window System
x11	6003/tcp	X Window System

x11	6003/udp	X Window System
x11	6004/udp	X Window System
x11	6004/tcp	X Window System
x11	6005/tcp	X Window System
x11	6005/udp	X Window System
x11	6006/tcp	X Window System | Bad Blood Trojan
x11	6006/udp	X Window System
x11	6007/udp	X Window System
x11	6007/tcp	X Window System
x11	6008/udp	X Window System
x11	6008/tcp	X Window System
x11	6009/tcp	X Window System
x11	6009/udp	X Window System
x11	6010/tcp	X Window System
x11	6010/udp	X Window System
x11	6011/udp	X Window System
x11	6011/tcp	X Window System
x11	6012/tcp	X Window System
x11	6012/udp	X Window System
x11	6013/udp	X Window System
x11	6013/tcp	X Window System
x11	6014/tcp	X Window System
x11	6014/udp	X Window System
x11	6015/tcp	X Window System
x11	6015/udp	X Window System
x11	6063/tcp	X Window System
x11	6063/udp	X Window System
irc	6667/tcp	Internet Relay Chat | SubSeven Trojan | NetBus Trojan
irc	6668/tcp	Internet Relay Chat
portmap	10000/udp	portmap portmappper sunrpc rpcbind | Network Data Management Protocol
rstatd	10001/udp	rpc rstatd
rstatd	10001/tcp	rpc rstatd
rusersd	10002/tcp	rpc rusersd
rusersd	10002/udp	rpc rusersd
nfsprog	10003/tcp	rpc nfsprog
nfsprog	10003/udp	rpc nfsprog
mountd	10005/tcp	mountd mount showmount | OpWin Trojan | Secure telnet

ypbind	10007/udp	ypbind | MVS Capacity
ypbind	10007/tcp	ypbind | MVS Capacity
yppasswdd	10009/tcp	NIS - yppasswd
yppasswdd	10009/udp	NIS - yppasswd
nlockmgr	10021/tcp	nfs lock manager
nlockmgr	10021/udp	nfs lock manager
rpc_statd	10024/udp	rpc status daemon
rpc_statd	10024/tcp	rpc status daemon
bootparam	10026/tcp	bootparam
bootparam	10026/udp	bootparam
pgpkeyserv	11371/tcp	PGP key server
pgpkeyserv	11371/udp	PGP key server
h323callsigalt	11720/udp	h323 Call Signal Alternate
h323callsigalt	11720/tcp	h323 Call Signal Alternate
trojan	16660/tcp	Stacheldraht Trojan
trojan	27665/tcp	Trinoo Trojan Master port
trojan	31335/udp	Trinoo trojan slave to master
trojan	34555/udp	trinoo trojan
trojan	35555/udp	trinoo trojan

Anexo C. *Netfilter:* Filtro de Redes para Linux

O *Netfilter* (filtro de rede) é uma estrutura para tratamento de pacotes, fora da estrutura de *socket Berkeley* normal, para o Sistema Operacional Linux. Ele está dividido em quatro partes, sendo que as três primeiras tratam do pacote ainda quando o cerne do Sistema Operacional está fazendo o tratamento das rotas dos pacotes e a última parte consiste em APIs para futuro desenvolvimento ou tratamentos especiais.

Na primeira parte são definidos “ganchos” em cada protoloco de rede. Estes “ganchos” são os pontos de passagem nos estágios dos protocolos. Em cada passagem de um estágio a outro é chamada a estrutura de tratamento do *netfilter* e conjuntamente é passado o número do “gancho”.

Na segunda parte, partes do cerne podem registrar seus pedidos para diferentes ganchos de cada protocolo. Então, a cada passagem de um estágio a outro, o cerne verifica se existe algum processo que tenha se registrado para analisar a passagem de um ponto a outro. O processo que está registrado tem a possibilidade de examinar (ou até mesmo alterar) o pacote e então pode: [1] descartar (NF_DROP); [2] permitir sua passagem (NF_ACCEPT); pedir ao *netfilter* para “esquecer” o pacote (NF_STOLEN); ou requisitar que o pacote seja colocado numa fila denominada *userspace* (NF_QUEUE) para outros tratamentos.

Na terceira parte, os pacotes que foram marcados para ir para a fila são coletados (pelo *driver ip_queue*) e mandados para o *userspace*. Os pacotes são dirigidos sem sincronia.

Os ganchos definidos variam de protocolo a protoloco, no caso do IP versão 4 (IPv4), são cinco ganchos:

1º - É onde os pacotes entram. Depois de analisado todos os controles de erro normais dos protocolos (soma de verificação, fragmentação, etc.), os pacotes são enviados para a estrutura (regras) de pré-rota (NF_IP_PRE_ROUTING) do *netfilter*.

2º - Se o pacote é destinado à máquina onde ele se encontra, a estrutura do *netfilter* é chamada para verificação do pacote (NF_IP_LOCAL_IN) antes de ser enviado ao processo local.

3º - Se o pacote tem como destino outro dispositivo de rede, a estrutura do *netfilter* é chamada antes da passagem de pacotes (NF_IP_FORWARD).

4º - Após ter passado pela tabela de rotas do Sistema Operacional, os pacotes são submetidos à estrutura do *netfilter* (NF_IP_POST_ROUTING) antes de efetivamente serem enviados.

5º - Este gancho (NF_IP_LOCAL_OUT) está definido para os pacotes de saída criados localmente na máquina. É chamado antes de ser passado para a tabela de rotas do cerne do Sistema Operacional.

A seleção dos pacotes para a filtragem é feita através da chamada *IP Tables*. O método de seleção dos pacotes é usado para [1] filtro de pacotes, [2] tradução de endereços de rede (NAT), [3] funções de pré-rota e [4] pós-rota de pacotes.

Anexo D. Snort: Sistema de Detecção de Intrusos

O Snort é um sistema de detecção de intrusos baseado em arquitetura centralizada, dados coletados na rede, análise baseada em assinaturas e com respostas ativas e passivas. Criado por Marty Roesh, faz análise em tempo real, passando todos os pacotes capturados na rede para um processador de regras. As regras são simples e parecidas com regras de filtro de pacotes.

```
alert tcp any any -> 200.223.45.2 80 (msg: "Ataque Code Red II"; flags: A+;
 uricontent:"scripts/root.exe?"; nocase; classtype: web-application-attack;
 sid: 1256; rev:2;)
```

Exemplo de regra do IDS Snort.

A primeira parte de uma regra define a ação a ser tomada. As opções são [1] *alert*, gera um alerta conforme o método de alerta (manda mensagem em rede Windows, mensagem em formato XML para uma outra máquina, envia e-mail para administrador, etc.) e depois escreve em arquivo de registro do sistema; [2] *log*, registra o pacote; [3] *pass*, ignora o pacote; [4] *activate*, alerta e ativa outra regra dinâmica; e [5] *dynamic*, regra específica que só realiza o serviço quando ativada, funciona como uma função de *log*.

A segunda parte da regra define o padrão a ser procurado, que pode ser apenas informações no cabeçalho IP, como também informações específicas dentro do pacote, como uma seqüência hexadecimal, mensagem de chamada de comando (ex: *"scrips/root.exe"*), bandeiras (*flags*) de protocolos, etc.

A partir da versão 1.5, foi acrescentado ao Snort a funcionalidade do uso de pré-processadores, que são uma espécie de "*plugins*" para análise de aspectos específicos como varredores de portas, re-organizadores de fragmentos IP, etc. Os pré-processadores são utilizados antes do pacote ser analisado, logo após ser capturado na rede.

Anexo E. Código fonte das regras dos *Screening Routers* e do IDS

Código do *Screening Router* interno:

```
#!/bin/sh


######### Configurações para o Screening Router Interno #########


###### Configurando kernel ######


# Ignora qualquer requisição de resposta a pedidos ICMP em difusão:
# defesa da rede contra ataques SMURF (Capítulo 3, Seção 3.6).
if [ -e /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts ]; then
       echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
fi

# Habilita proteção contra mensagens de erros falsos: negação de
# serviços
# (Capítulo 3, Seção 3.6).
if [ -e /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses ]; then
       echo "0" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
fi

# Escreve no arquivo de log do kernel pacotes com endereços
# impossíveis: falsificação de endereços IP e negação de serviços
#(Capítulo 3, seções 3.5 e 3.6).
if [ -e /proc/sys/net/ipv4/conf/all/log_martians ]; then
       echo "1" > /proc/sys/net/ipv4/conf/all/log_martians
fi

# Taxas ICMP é a quantidade de respostas num determinado tempo em
# milissengundos. Evita o firewall seja usado como amplificadores num
# ataque de negação de serviços e que ele próprio seja vítima.

# Taxas ICMP
if [ -e /proc/sys/net/ipv4/icmp_echoreply_rate ]; then
       echo "200" > /proc/sys/net/ipv4/icmp_echoreply_rate
fi

# Taxas ICMP
if [ -e /proc/sys/net/ipv4/icmp_paramprob_rate ]; then
       echo "200" > /proc/sys/net/ipv4/icmp_paramprob_rate
fi

# Taxas ICMP
if [ -e /proc/sys/net/ipv4/icmp_timeexceed_rate ]; then
       echo "200" > /proc/sys/net/ipv4/icmp_timeexceed_rate
fi

# Taxas ICMP
if [ -e /proc/sys/net/ipv4/icmp_destunreach_rate ]; then
       echo "200" > /proc/sys/net/ipv4/icmp_destunreach_rate
fi
```

```
# TCP SYN cookies : proteção contra IP com origem falsas (Capítulo 3,
# Seção 3.5).
if [ -e /proc/sys/net/ipv4/tcp_syncookies ]; then
       echo "1" > /proc/sys/net/ipv4/tcp_syncookies
fi

# Habilita repasse de pacotes IP
if [ -e /proc/sys/net/ipv4/ip_forward ]; then
       echo "1" > /proc/sys/net/ipv4/ip_forward
fi


###### Declaração de variáveis ######


IPTABLES="/usr/sbin/iptables"      ## Caminho do iptables

LOOPBACK="lo"                      ## Interface de Loopback
FR_RELAY="eth0"                    ## Interface da ligação com o roteador FR_RELAY
INTERNA="eth1"                     ## Interface da rede interna
BACKBONE="eth2"                    ## Interface com o Backbone Interno

REDE_LOOPBACK="127.0.0/8"          ## Endereço de loopback
REDE_FR_RELAY="10.100.1.0/30"      ## Endereço da rede com o roteador FR_RELAY
REDE_INTERNA="10.20.1.0/24"        ## Endereço da rede interna
REDE_BACKBONE="10.1.1.0/30"      ## Endereço da rede com o Backbone

IP_LOOPBACK="127.0.0.1"            ## Endereço IP de loopback
IP_FR_RELAY="10.100.1.1"           ## Endereço IP da placa com o
                                          ## roteador FR_RELAY
IP_INTERNA="10.20.1.1"           ## Endereço IP da placa de rede interna
IP_BACKBONE="10.1.1.2"                ## Endereço IP da placa com o Backbone

REDE_INTERNET="200.254.130.0/26"   ## Endereço da rede de Servidores
                                          ## Internet ALFA

SERVIDOR_MAIL="200.254.130.10"   ## Endereço do Servidor de e-mail
SERVIDOR_HTTP_1="200.254.130.2"  ## Endereço do Servidor de http no. 1
SERVIDOR_HTTP_2="200.254.130.10" ## Endereço do Servidor de http no. 2
SERVIDOR_DNS_1="200.254.130.10"  ## Endereço do Servidor de dns no. 1
SERVIDOR_DNS_2="200.254.130.2"   ## Endereço do Servidor de dns no. 2
SERVIDOR_BETA="200.254.130.25"   ## Endereço do Servidor do BETA
SERVIDOR_BD="10.20.1.4"            ## Endereço do Servidor de Banco de Dados
FIREWALL_INTERNET="10.1.1.1"        ## Endereço do firewall internet
ROTEADOR_FR_RELAY="10.100.1.2"     ## Endereço do roteador FR_RELAY
SERVIDOR_BD_GAMA="172.30.0.6"          ## Endereço do Servidor de BD do GAMA


###### Regras ######


## Flush/Destroi regras ##

$IPTABLES -F INPUT
$IPTABLES -F OUTPUT
$IPTABLES -F FORWARD
$IPTABLES -F -t nat
$IPTABLES -F -t mangle
$IPTABLES -F ICMP_AMIGA
```

```
$IPTABLES -F ICMP_SUSPEITA
$IPTABLES -F FINALIZA_FLOOD

## Destroi regras definidas pelo usuário ##

$IPTABLES -X

## Criando políticas padrão ##
# O Screening Router Interno tem a negação como política padrão
# Capítulo 3, Seção 1.5

$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP


# AS correntes de regras ICMP para redes amigas e suspeitas são
# correntes de validação que um tráfego ICMP é submetido. São
# construídas duas correntes: [1] Amigas, para redes internas e
# confiáveis; [2] Suspeitas, para redes externas e não confiáveis.


###### Regras de ICMP para redes amigas ######


#### Criando Corrente de regras ####
$IPTABLES -N ICMP_AMIGA
$IPTABLES -F ICMP_AMIGA

#### Regras propriamente ditas ####
## Ecoa o ping ##
$IPTABLES -A ICMP_AMIGA -p icmp --icmp-type echo-reply -j ACCEPT

## Destino inalcançável ##
$IPTABLES -A ICMP_AMIGA -p icmp --icmp-type destination-unreachable -j ACCEPT

## Diminuição no tráfego ##
$IPTABLES -A ICMP_AMIGA -p icmp --icmp-type source-quench -j ACCEPT

## Tempo Excedido (traceroutes) ##
$IPTABLES -A ICMP_AMIGA -p icmp --icmp-type time-exceeded -j ACCEPT

## Problemas de parâmetros ##
$IPTABLES -A ICMP_AMIGA -p icmp --icmp-type parameter-problem -j ACCEPT


###### Regras de ICMP para redes suspeitas ######


#### Criando Corrente de regras ####
$IPTABLES -N ICMP_SUSPEITAS
$IPTABLES -F ICMP_SUSPEITAS



#### Regras propriamente ditas ####

## Diminuição no tráfego ##
$IPTABLES -A ICMP_SUSPEITAS -p icmp --icmp-type source-quench -j ACCEPT
```

```
## Requisita resposta (eco) (ping) ##
$IPTABLES -A ICMP_SUSPEITAS -p icmp --icmp-type echo-request -j ACCEPT

## Problemas de parâmetros ##
$IPTABLES -A ICMP_SUSPEITAS -p icmp --icmp-type parameter-problem -j ACCEPT


###### Regras de pré roteamento ######


#### Regras para ajustes no TOS (Tipo de Serviço) (Capítulo 2, Seção 1.1) ####

## Diminiu atraso para acesso SSH (bit de atraso) ##
$IPTABLES -A PREROUTING -t mangle -p tcp --dport ssh -j TOS --set-tos Minimize-Delay

# As duas regras principais para finalizar e registrar a falsificação
# de endereços IP de origem são: [1] endereço IP reservado a redes
# internas vindo em interfaces de redes externas (internet) é
# falsificação; [2] endereço IP correto (válido e que é da rede) porém
# vindo da interface errada é falsificação.


###### Regras para finalização de IP Spoofing (falsificação de IP de
###### origem) (Capítulo 3, Seção 3.5) ######


#### Criando Corrente de regras ####
$IPTABLES -N FINALIZA_SPOOF
$IPTABLES -F FINALIZA_SPOOF

#### Regras propriamente ditas ####

## Escreve tentativa no log do sistema (syslog) ##
$IPTABLES -A FINALIZA_SPOOF -m limit --limit 1/h --limit-burst 5 -j LOG --log-prefix
"Firewall: Spoof! "

## Descarta pacote ##
$IPTABLES -A FINALIZA_SPOOF -j DROP


# A regra principal para finalizar e registrar a negação de serviço é:
# a verificação de assinaturas de DoS pelos bits IP.


###### Regras para finalização de IP Flooding (tentativa de negação de
###### serviço)  (Capítulo 3, Seção 3.6) ######



#### Criando Corrente de regras ####
$IPTABLES -N FINALIZA_FLOOD
$IPTABLES -F FINALIZA_FLOOD

#### Regras propriamente ditas ####

## Escreve tentativa no log do sistema (syslog) ##
```

```
$IPTABLES -A FINALIZA_FLOOD -m limit --limit 1/h --limit-burst 5 -j LOG --log-prefix
"Firewall: Flood! "

## Descarta pacote ##
$IPTABLES -A FINALIZA_FLOOD -j DROP


###### Regras para a rede loopback ######


#### Regras para entrada vinda do loopback ####
$IPTABLES -A INPUT -i $LOOPBACK -j ACCEPT

#### Regras para saida do localhost (maquina local) ####
$IPTABLES -A OUTPUT -o $LOOPBACK -j ACCEPT


###### Regras para entradas (INPUT) ######


#### Verifica tentativas de negação de serviços (Capítulo 3, Seção 3.6) ####
$IPTABLES -A INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -j FINALIZA_FLOOD

$IPTABLES -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j FINALIZA_FLOOD

$IPTABLES -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j FINALIZA_FLOOD

$IPTABLES -A INPUT -p tcp --tcp-flags ALL NONE -j FINALIZA_FLOOD


#### Regras para entrada vinda da "Nuvem" FR_RELAY ####

## Verifica tentativas de ataques ##
# Falsificação de endereço IP de origem no pacote (Capítulo 3, Seção
# 3.5) #
$IPTABLES -A INPUT -i $FR_RELAY -s $REDE_INTERNA -j FINALIZA_SPOOF

## ICMP ##
$IPTABLES -A INPUT -i $FR_RELAY -d $IP_FR_RELAY -p icmp -j ICMP_SUSPEITA

## Tráfego Normal ##
$IPTABLES -A INPUT -i $FR_RELAY -m state --state RELATED,ESTABLISHED -j ACCEPT


#### Regras para entrada vinda da Rede Interna ####
## ICMP ##
$IPTABLES -A INPUT -i $INTERNA -s $REDE_INTERNA -d $IP_INTERNA -p icmp -j ICMP_AMIGA

## Tráfego Normal ##
$IPTABLES -A INPUT -i $INTERNA -s $REDE_INTERNA -d $IP_INTERNA -m state --state
RELATED,ESTABLISHED -j ACCEPT

# Aceita conexões SSH #
$IPTABLES -A INPUT -i $INTERNA -s $REDE_INTERNA -d $IP_INTERNA -m state --state !
INVALID -p tcp --dport 22 -j ACCEPT


#### Regras para entrada vinda do Backbone ####
## ICMP ##
```

```
$IPTABLES -A INPUT -i $BACKBONE -d $IP_BACKBONE -p icmp -j ICMP_SUSPEITA

## Tráfego Normal ##
$IPTABLES -A INPUT -i $BACKBONE -d $IP_BACKBONE -m state --state RELATED,ESTABLISHED
-j ACCEPT


###### Regras de saídas (OUTPUT) ######


#### Regras de saída para a "Nuvem" FR_RELAY ####
## ICMP ##
$IPTABLES -A OUTPUT -o $FR_RELAY -s $IP_FR_RELAY -p icmp -j ICMP_AMIGA

## Tráfego Normal ##
$IPTABLES -A OUTPUT -o $FR_RELAY -s $IP_FR_RELAY -m state --state RELATED,ESTABLISHED
-j ACCEPT


#### Regras de saída para a Rede Interna ####
## ICMP ##
$IPTABLES -A OUTPUT -o $INTERNA -s $IP_INTERNA -d $REDE_INTERNA -p icmp -j ICMP_AMIGA

## Tráfego Normal ##
$IPTABLES -A OUTPUT -o $INTERNA -s $IP_INTERNA -d $REDE_INTERNA -m state --state !
INVALID -j ACCEPT


#### Regras de saída para o Backbone ####
## ICMP ##
$IPTABLES -A OUTPUT -o $BACKBONE -s $IP_BACKBONE -p icmp -j ICMP_AMIGA

## Tráfego Normal ##
$IPTABLES -A OUTPUT -o $BACKBONE -s $IP_BACKBONE -m state --state ! INVALID -j ACCEPT


###### Regras de repasses (FORWARD) ######


#### Verifica tentativas de negação de serviços (Capítulo 3, Seção 3.6) ####
$IPTABLES -A FORWARD -p tcp --tcp-flags ALL FIN,URG,PSH -j FINALIZA_FLOOD
$IPTABLES -A FORWARD -p tcp --tcp-flags SYN,RST SYN,RST -j FINALIZA_FLOOD
$IPTABLES -A FORWARD -p tcp --tcp-flags SYN,FIN SYN,FIN -j FINALIZA_FLOOD
$IPTABLES -A FORWARD -p tcp --tcp-flags ALL NONE -j FINALIZA_FLOOD


#### Regras de repasses para a "Nuvem" FR_RELAY ####
## Verifica tentativas de ataques ##
# Falsificação de endereço IP de origem no pacote (Capítulo 3, Seção 3.5) #
$IPTABLES -A FORWARD -i $FR_RELAY -s $REDE_INTERNA -j FINALIZA_SPOOF

## ICMP ##
# Rede Interna pelo FR_RELAY  #
$IPTABLES -A FORWARD -i $INTERNA -o $FR_RELAY -s $REDE_INTERNA -p icmp -j ICMP_AMIGA

## Tráfego Normal ##
# Conexão ao Servidor de Banco de Dados GAMA #
$IPTABLES -A FORWARD -i $INTERNA -o $FR_RELAY -d $SERVIDOR_BD_GAMA -m state --state !
INVALID -p tcp --dport 5000 -j ACCEPT
```

```
#### Regras de repasses para a Rede Interna ####
## ICMP ##
# FR_RELAY pela Rede Interna #
$IPTABLES -A FORWARD -i $FR_RELAY -o $INTERNA -d $REDE_INTERNA -p icmp -j
ICMP_SUSPEITA

# Backbone pela Rede Interna #
$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -d $REDE_INTERNA -p icmp -j
ICMP_SUSPEITA

## Tráfego Normal ##
# Responda da conexão ao Servidor de Banco do Dados GAMA #
$IPTABLES -A FORWARD -i $FR_RELAY -o $INTERNA -s $SERVIDOR_BD_GAMA -d $REDE_INTERNA -
m state --state RELATED,ESTABLISHED -p tcp --sport 5000 -j ACCEPT

# Servidores Internet #
$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -s $SERVIDOR_MAIL -d $REDE_INTERNA -m
state --state RELATED,ESTABLISHED -p tcp --sport 25 -j ACCEPT

$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -s $SERVIDOR_MAIL -d $REDE_INTERNA -m
state --state RELATED,ESTABLISHED -p tcp --sport 110 -j ACCEPT

$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -s $SERVIDOR_HTTP_1 -d $REDE_INTERNA -m
state --state RELATED,ESTABLISHED -p tcp --sport 20 -j ACCEPT

$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -s $SERVIDOR_HTTP_1 -d $REDE_INTERNA -m
state --state RELATED,ESTABLISHED -p tcp --sport 21 -j ACCEPT

$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -s $SERVIDOR_HTTP_2 -d $REDE_INTERNA -m
state --state RELATED,ESTABLISHED -p tcp --sport 20 -j ACCEPT

$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -s $SERVIDOR_HTTP_2 -d $REDE_INTERNA -m
state --state RELATED,ESTABLISHED -p tcp --sport 21 -j ACCEPT

$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -s $SERVIDOR_DNS_1 -d $REDE_INTERNA -m
state --state RELATED,ESTABLISHED -p udp --sport 53 -j ACCEPT

$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -s $SERVIDOR_DNS_2 -d $REDE_INTERNA -m
state --state RELATED,ESTABLISHED -p udp --sport 53 -j ACCEPT

# Serviços de banco de dados dos servidores internet Alfa #
$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -s $SERVIDOR_HTTP_2 -d $SERVIDOR_BD -m
state --state ! INVALID -p tcp --dport 5432 -j ACCEPT

# Conexões Internet #
$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -d $REDE_INTERNA -m state --state
RELATED,ESTABLISHED -p udp --sport 80 -j ACCEPT

$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -d $REDE_INTERNA -m state --state
RELATED,ESTABLISHED -p tcp --sport 80 -j ACCEPT
$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -d $REDE_INTERNA -m state --state
RELATED,ESTABLISHED -p tcp --sport 443 -j ACCEPT

$IPTABLES -A FORWARD -i $BACKBONE -o $INTERNA -d $REDE_INTERNA -m state --state
RELATED,ESTABLISHED -p tcp --sport 443 -j ACCEPT
```

```
#### Regras de repasses para o Backbone ####
## ICMP ##
# Rede Interna pelo Backbone #
$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $REDE_INTERNA -p icmp -j ICMP_AMIGA

## Tráfego Normal ##
# Resposta do bando de dados à servidores internet Alfa #
$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $SERVIDOR_BD -d $SERVIDOR_HTTP_2 -m
state --state RELATED,ESTABLISHED -p tcp --sport 5432 -j ACCEPT

# Liberando acesso para máquinas acessarem a internet #
CLASSE_REDE="10.20.1"
TERMINADOR=20

while [ "$TERMINADOR" -le 99 ]; do

      $IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $CLASSE_REDE.$TERMINADOR -m
state --state ! INVALID -d ! $REDE_INTERNET -p tcp --dport 80 -j ACCEPT

       $IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $CLASSE_REDE.$TERMINADOR -m
state --state ! INVALID -d ! $REDE_INTERNET -p udp --dport 80 -j ACCEPT

       $IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $CLASSE_REDE.$TERMINADOR -m
state --state ! INVALID -d ! $REDE_INTERNET -p tcp --dport 443 -j ACCEPT

       $IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $CLASSE_REDE.$TERMINADOR -m
state --state ! INVALID -d ! $REDE_INTERNET -p udp --dport 443 -j ACCEPT

      TERMINADOR=$(($TERMINADOR+1))
done

# Acesso aos servidores internet da Alfa #
$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $REDE_INTERNA -m state --state !
INVALID -d $SERVIDOR_MAIL -p tcp --dport 25 -j ACCEPT

$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $REDE_INTERNA -m state --state !
INVALID -d $SERVIDOR_MAIL -p tcp --dport 110 -j ACCEPT

$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $REDE_INTERNA -m state --state !
INVALID -d $SERVIDOR_DNS_1 -p udp --dport 53 -j ACCEPT

$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $REDE_INTERNA -m state --state !
INVALID -d $SERVIDOR_DNS_2 -p udp --dport 53 -j ACCEPT

$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $REDE_INTERNA -m state --state !
INVALID -d $SERVIDOR_HTTP_1 -p tcp --dport 80 -j ACCEPT

$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $REDE_INTERNA -m state --state !
INVALID -d $SERVIDOR_HTTP_1 -p udp --dport 80 -j ACCEPT

$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $REDE_INTERNA -m state --state !
INVALID -d $SERVIDOR_HTTP_1 -p tcp --dport 443 -j ACCEPT

$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $REDE_INTERNA -m state --state !
INVALID -d $SERVIDOR_HTTP_1 -p udp --dport 443 -j ACCEPT

$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $REDE_INTERNA -m state --state !
INVALID -d $SERVIDOR_HTTP_2 -p tcp --dport 80 -j ACCEPT
```

```
$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $REDE_INTERNA -m state --state !
INVALID -d $SERVIDOR_HTTP_2 -p udp --dport 80 -j ACCEPT

$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $REDE_INTERNA -m state --state !
INVALID -d $SERVIDOR_BETA -p tcp --dport 80 -j ACCEPT

$IPTABLES -A FORWARD -i $INTERNA -o $BACKBONE -s $REDE_INTERNA -m state --state !
INVALID -d $SERVIDOR_BETA -p udp --dport 80 -j ACCEPT
```

Código do *Screening Router* externo:

```
#!/bin/sh


######## Configurações para o Screening Router Interno ########


###### Configurando kernel ######


# Ignora qualquer requisição de resposta a pedidos ICMP em difusão: defesa da
# rede contra ataques SMURF (Capítulo 3, Seção 3.6).
if [ -e /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts ]; then
        echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
fi

# Habilita proteção contra mensagens de erros falsos: negação de serviços
# (Capítulo 3, Seção 3.6).
if [ -e /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses ]; then
        echo "0" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
fi

# Escreve no arquivo de log do kernel pacotes com endereços impossíveis:
# falsificação de endereços IP e negação de serviços (Capítulo 3, seções 3.5 e 3.6).
if [ -e /proc/sys/net/ipv4/conf/all/log_martians ]; then
        echo "1" > /proc/sys/net/ipv4/conf/all/log_martians
fi

# Habilita opções para conexões com endereços IP dinâmicos (Point to Point
# Protocol - PPP)
if [ -e /proc/sys/net/ipv4/ip_dynaddr ]; then
        echo "1" > /proc/sys/net/ipv4/ip_dynaddr
fi

# Taxas ICMP é a quantidade de respostas num determinado tempo em
# milissengundos. Evita o firewall seja usado como amplificadores num ataque
# de negação de serviços e que ele próprio seja vítima.

# Taxas ICMP
if [ -e /proc/sys/net/ipv4/icmp_echoreply_rate ]; then
        echo "200" > /proc/sys/net/ipv4/icmp_echoreply_rate
fi

# Taxas ICMP
if [ -e /proc/sys/net/ipv4/icmp_paramprob_rate ]; then
        echo "200" > /proc/sys/net/ipv4/icmp_paramprob_rate
fi

# Taxas ICMP
if [ -e /proc/sys/net/ipv4/icmp_timeexceed_rate ]; then
        echo "200" > /proc/sys/net/ipv4/icmp_timeexceed_rate
fi

# Taxas ICMP
if [ -e /proc/sys/net/ipv4/icmp_destunreach_rate ]; then
        echo "200" > /proc/sys/net/ipv4/icmp_destunreach_rate
fi
```

```
# TCP SYN cookies : proteção contra IP com origem falsas (Capítulo 3,
# Seção 3.5).
if [ -e /proc/sys/net/ipv4/tcp_syncookies ]; then
        echo "1" > /proc/sys/net/ipv4/tcp_syncookies
fi

# Habilita repasse de pacotes IP
if [ -e /proc/sys/net/ipv4/ip_forward ]; then
        echo "1" > /proc/sys/net/ipv4/ip_forward
fi


###### Declaração de variáveis ######

IPTABLES="/usr/sbin/iptables"           ## Caminho do iptables

LOOPBACK="lo"                           ## Interface de Loopback
LINK="eth0"                 ## Interface da ligação com o roteador internet
SERVIDORES="eth1"               ## Interface da rede de servidores internet
BACKBONE="eth2"                         ## Interface com o Backbone Interno
ADSL="ppp0"                        ## Interface com o modem ADSL

REDE_LOOPBACK="127.0.0/8"               ## Endereço de loopback
REDE_LINK="200.254.130.252/30"      ## Endereço da rede com o roteador internet
REDE_SERVIDORES="200.254.130.0/26"  ## Endereço da rede com servidores internet
REDE_BACKBONE="10.1.1.0/30"         ## Endereço da rede com o Backbone

IP_LOOPBACK="127.0.0.1"                 ## Endereço IP de loopback
IP_LINK="200.254.130.253"        ## Endereço IP da placa com o roteador internet
IP_SERVIDORES="200.254.130.1" ## Endereço IP da placa com os servidores internet
IP_BACKBONE="10.1.1.1"                    ## Endereço IP da placa com o Backbone

SERVIDOR_MAIL="200.254.130.10"          ## Endereço do Servidor de e-mail
SERVIDOR_HTTP_1="200.254.130.2"         ## Endereço do Servidor de http no. 1
SERVIDOR_HTTP_2="200.254.130.10"        ## Endereço do Servidor de http no. 2
SERVIDOR_DNS_1="200.254.130.10"         ## Endereço do Servidor de dns no. 1
SERVIDOR_DNS_2="200.254.130.2"          ## Endereço do Servidor de dns no. 2
SERVIDOR_BETA="200.254.130.25"            ## Endereço do Servidor do BETA
SERVIDOR_BD="10.20.1.4"                   ## Endereço do Servidor Banco de Dados

REDE_INTERNA="10.20.1.0/24"        ## Endereço da rede interna da Alfa

REDES_RESERVADAS="0.0.0.0/8 1.0.0.0/8 2.0.0.0/8 5.0.0.0/8 10.0.0.0/8 \
        23.0.0.0/8 27.0.0.0/8 31.0.0.0/8 36.0.0.0/8 37.0.0.0/8 \
        39.0.0.0/8 41.0.0.0/8 42.0.0.0/8 58.0.0.0/8 59.0.0.0/8 \
        60.0.0.0/8 69.0.0.0/8 70.0.0.0/8 71.0.0.0/8 72.0.0.0/8 \
        73.0.0.0/8 74.0.0.0/8 75.0.0.0/8 76.0.0.0/8 77.0.0.0/8 \
        78.0.0.0/8 79.0.0.0/8 82.0.0.0/8 83.0.0.0/8 84.0.0.0/8 85.0.0.0/8 \
        86.0.0.0/8 87.0.0.0/8 88.0.0.0/8 89.0.0.0/8 90.0.0.0/8 91.0.0.0/8 \
        92.0.0.0/8 93.0.0.0/8 94.0.0.0/8 95.0.0.0/8 96.0.0.0/8 97.0.0.0/8 \
        98.0.0.0/8 99.0.0.0/8 100.0.0.0/8 101.0.0.0/8 102.0.0.0/8 \
        103.0.0.0/8 104.0.0.0/8 105.0.0.0/8 106.0.0.0/8 107.0.0.0/8 \
        108.0.0.0/8 109.0.0.0/8 110.0.0.0/8 111.0.0.0/8 112.0.0.0/8 \
        113.0.0.0/8 114.0.0.0/8 115.0.0.0/8 116.0.0.0/8 117.0.0.0/8 \
        118.0.0.0/8 119.0.0.0/8 120.0.0.0/8 121.0.0.0/8 122.0.0.0/8 \
        123.0.0.0/8 124.0.0.0/8 125.0.0.0/8 126.0.0.0/8 127.0.0.0/8 \
        128.0.0.0/16 172.16.0.0/12 191.255.0.0/16 192.0.0.0/16 192.168.0.0/16 \
        197.0.0.0/8 201.0.0.0/8 219.0.0.0/8 220.0.0.0/8 221.0.0.0/8 \
```

```
        222.0.0.0/8 223.0.0.0/8 240.0.0.0/8 241.0.0.0/8 242.0.0.0/8 \
        243.0.0.0/8 244.0.0.0/8 245.0.0.0/8 246.0.0.0/8 247.0.0.0/8 \
        248.0.0.0/8 249.0.0.0/8 250.0.0.0/8 251.0.0.0/8 252.0.0.0/8 \
        253.0.0.0/8 254.0.0.0/8 255.0.0.0/8"


###### Regras ######


## Flush/Destroi regras ##

$IPTABLES -F INPUT
$IPTABLES -F OUTPUT
$IPTABLES -F FORWARD
$IPTABLES -F -t nat
$IPTABLES -F -t mangle
$IPTABLES -F ICMP_AMIGA
$IPTABLES -F ICMP_SUSPEITA
$IPTABLES -F FINALIZA_SPOOF
$IPTABLES -F FINALIZA_FLOOD

## Destroi regras definidas pelo usuário ##

$IPTABLES -X

## Criando políticas padrão ##
# O Screening Router Externo tem a negação como política padrão
# Capítulo 3, Seção 1.5

$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP


# AS correntes de regras ICMP para redes amigas e suspeitas são correntes
# de validação que um tráfego ICMP é submetido. São construídas duas
# correntes: [1] Amigas, para redes internas e confiáveis; [2] Suspeitas,
# para redes externas e não confiáveis.

###### Regras de ICMP para redes amigas ######


#### Criando Corrente de regras ####
$IPTABLES -N ICMP_AMIGA
$IPTABLES -F ICMP_AMIGA

#### Regras propriamente ditas ####
## Ecoa o ping ##
$IPTABLES -A ICMP_AMIGA -p icmp --icmp-type echo-reply -j ACCEPT

## Destino inalcançável ##
$IPTABLES -A ICMP_AMIGA -p icmp --icmp-type destination-unreachable -j ACCEPT


## Diminuição no tráfego ##
$IPTABLES -A ICMP_AMIGA -p icmp --icmp-type source-quench -j ACCEPT

## Tempo Excedido (traceroutes) ##
$IPTABLES -A ICMP_AMIGA -p icmp --icmp-type time-exceeded -j ACCEPT
```

```
## Problemas de parâmetros ##
$IPTABLES -A ICMP_AMIGA -p icmp --icmp-type parameter-problem -j ACCEPT


###### Regras de ICMP para redes suspeitas ######


#### Criando Corrente de regras ####
$IPTABLES -N ICMP_SUSPEITAS
$IPTABLES -F ICMP_SUSPEITAS

#### Regras propriamente ditas ####
## Diminuição no tráfego ##
$IPTABLES -A ICMP_SUSPEITAS -p icmp --icmp-type source-quench -j ACCEPT

## Requisita resposta (eco) (ping) ##
$IPTABLES -A ICMP_SUSPEITAS -p icmp --icmp-type echo-request -j ACCEPT

## Problemas de parâmetros ##
$IPTABLES -A ICMP_SUSPEITAS -p icmp --icmp-type parameter-problem -j ACCEPT


###### Regras de pré roteamento ######


#### Regras para ajustes no TOS (Tipo de Serviço) (Capítulo 2, Seção 1.1) ####
## Diminiu atraso para acesso SSH (bit de atraso) ##
$IPTABLES -A PREROUTING -t mangle -p tcp --dport ssh -j TOS --set-tos Minimize-Delay

## Melhoria na performance de acesso aos servidores HTTP (bit de passagem rápida) ##
$IPTABLES -A PREROUTING -t mangle -p tcp -s $SERVIDOR_HTTP_1 --sport 80 -j TOS --set-
tos Maximize-throughput
$IPTABLES -A PREROUTING -t mangle -p tcp -s $SERVIDOR_HTTP_1 --sport 443 -j TOS --
set-tos Maximize-throughput
$IPTABLES -A PREROUTING -t mangle -p tcp -s $SERVIDOR_HTTP_2 --sport 80 -j TOS --set-
tos Maximize-throughput
$IPTABLES -A PREROUTING -t mangle -p tcp -s $SERVIDOR_BETA --sport 80 -j TOS --set-
tos Maximize-throughput


# As duas regras principais para finalizar e registrar a falsificação de
# endereços IP de origem são: [1] endereço IP reservado a redes internas vindo
# em interfaces de redes externas (internet) é falsificação; [2] endereço IP
# correto (válido e que é da rede) porém vindo da interface errada é
# falsificação.


###### Regras para finalização de IP Spoofing (falsificação de IP de origem)
(Capítulo 3, Seção 3.5) ######


#### Criando Corrente de regras ####
$IPTABLES -N FINALIZA_SPOOF
$IPTABLES -F FINALIZA_SPOOF

#### Regras propriamente ditas ####
## Escreve tentativa no log do sistema (syslog) ##
```

```
$IPTABLES -A FINALIZA_SPOOF -m limit --limit 1/h --limit-burst 5 -j LOG --log-prefix
"Firewall: Spoof! "

## Descarta pacote ##
$IPTABLES -A FINALIZA_SPOOF -j DROP


# A regra principal para finalizar e registrar a negação de serviço é: a
# verificação de assinaturas de DoS pelos bits IP.


###### Regras para finalização de IP Flooding (tentativa de negação de serviço)
(Capítulo 3, Seção 3.6) ######


#### Criando Corrente de regras ####
$IPTABLES -N FINALIZA_FLOOD
$IPTABLES -F FINALIZA_FLOOD

#### Regras propriamente ditas ####
## Escreve tentativa no log do sistema (syslog) ##
$IPTABLES -A FINALIZA_FLOOD -m limit --limit 1/h --limit-burst 5 -j LOG --log-prefix
"Firewall: Flood! "

## Descarta pacote ##
$IPTABLES -A FINALIZA_FLOOD -j DROP


###### Regras para a rede loopback ######


#### Regras para entrada vinda do loopback ####
$IPTABLES -A INPUT -i $LOOPBACK -j ACCEPT

#### Regras para saida do localhost (maquina local) ####
$IPTABLES -A OUTPUT -o $LOOPBACK -j ACCEPT


###### Regras para entradas (INPUT) ######

#### Verifica tentativas de negação de serviços (Capítulo 3, Seção 3.6) ####
$IPTABLES -A INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -j FINALIZA_FLOOD
$IPTABLES -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j FINALIZA_FLOOD
$IPTABLES -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j FINALIZA_FLOOD
$IPTABLES -A INPUT -p tcp --tcp-flags ALL NONE -j FINALIZA_FLOOD

#### Regras para entrada vinda do Link Internet ####
## Verifica tentativas de ataques ##
# Falsificação de endereço IP de origem no pacote (Capítulo 3, Seção 3.5) #
for REDE in $REDES_RESERVADAS; do

       $IPTABLES -A INPUT -i $LINK -s $REDE -j FINALIZA_SPOOF

done

$IPTABLES -A INPUT -i $LINK -s $REDE_SERVIDORES -j FINALIZA_SPOOF

## ICMP ##
$IPTABLES -A INPUT -i $LINK -d $IP_LINK -p icmp -j ICMP_SUSPEITA
```

```
## Tráfego Normal ##
$IPTABLES -A INPUT -i $LINK -m state --state RELATED,ESTABLISHED -j ACCEPT


#### Regras para entrada vinda da Rede de Servidores ####
## Verifica tentativas de ataques ##
# Falsificação de endereço IP de origem no pacote (Capítulo 3, Seção 3.5) #
for REDE in $REDES_RESERVADAS; do

        $IPTABLES -A INPUT -i $SERVIDORES -s $REDE -j FINALIZA_SPOOF

done

$IPTABLES -A INPUT -i $SERVIDORES -s ! $REDE_SERVIDORES -j FINALIZA_SPOOF

## ICMP ##
$IPTABLES -A INPUT -i $SERVIDORES -s $REDE_SERVIDORES -d $IP_SERVIDORES -p icmp -j
ICMP_SUSPEITA

## Tráfego Normal ##
$IPTABLES -A INPUT -i $SERVIDORES -s $REDE_SERVIDORES -d $IP_SERVIDORES -m state --
state RELATED,ESTABLISHED -j ACCEPT


#### Regras para entrada vinda do Backbone ####
## ICMP ##
$IPTABLES -A INPUT -i $BACKBONE -d $IP_BACKBONE -p icmp -j ICMP_SUSPEITA

# Vindos da Rede Interna #
$IPTABLES -A INPUT -i $BACKBONE -s $REDE_INTERNA -d $IP_BACKBONE -p icmp -j
ICMP_AMIGA

## Tráfego Normal ##
$IPTABLES -A INPUT -i $BACKBONE -d $IP_BACKBONE -m state --state RELATED,ESTABLISHED
-j ACCEPT

# Aceita conexões SSH #
$IPTABLES -A INPUT -i $BACKBONE -s $REDE_INTERNA -d $IP_BACKBONE -m state --state !
INVALID -p tcp --dport 22 -j ACCEPT


#### Regras para entrada vinda do ADSL ####
## Verifica tentativas de ataques ##
# Falsificação de endereço IP de origem no pacote (Capítulo 3, Seção 3.5) #
for REDE in $REDES_RESERVADAS; do

        $IPTABLES -A INPUT -i $ADSL -s $REDE -j FINALIZA_SPOOF

done

$IPTABLES -A INPUT -i $ADSL -s $REDE_SERVIDORES -j FINALIZA_SPOOF

## Tráfego Normal ##
$IPTABLES -A INPUT -i $ADSL -m state --state RELATED,ESTABLISHED -j ACCEPT


###### Regras de saídas (OUTPUT) ######
```

```
#### Regras de saída para o Link Internet ####
## ICMP ##
$IPTABLES -A OUTPUT -o $LINK -s $IP_LINK -p icmp -j ICMP_AMIGA

## Tráfego Normal ##
$IPTABLES -A OUTPUT -o $LINK -s $IP_LINK -m state --state ! INVALID -j ACCEPT


#### Regras de saída para a Rede de Servidores ####
## ICMP ##
$IPTABLES -A OUTPUT -o $SERVIDORES -s $IP_SERVIDORES -d $REDE_SERVIDORES -p icmp -j
ICMP_AMIGA

## Tráfego Normal ##
$IPTABLES -A OUTPUT -o $SERVIDORES -s $IP_SERVIDORES -d $REDE_SERVIDORES -m state --
state ! INVALID -j ACCEPT


#### Regras de saída para o Backbone ####
## ICMP ##
$IPTABLES -A OUTPUT -o $BACKBONE -s $IP_BACKBONE -p icmp -j ICMP_AMIGA

## Tráfego Normal ##
$IPTABLES -A OUTPUT -o $BACKBONE -s $IP_BACKBONE -m state --state RELATED,ESTABLISHED
-j ACCEPT


#### Regras de saída para o ADSL ####
## ICMP ##
$IPTABLES -A OUTPUT -o $ADSL -p icmp -j ICMP_AMIGA


###### Regras de repasses (FORWARD) ######


#### Verifica tentativas de negação de serviços (Capítulo 3, Seção 3.6) ####
$IPTABLES -A FORWARD -p tcp --tcp-flags ALL FIN,URG,PSH -j FINALIZA_FLOOD
$IPTABLES -A FORWARD -p tcp --tcp-flags SYN,RST SYN,RST -j FINALIZA_FLOOD
$IPTABLES -A FORWARD -p tcp --tcp-flags SYN,FIN SYN,FIN -j FINALIZA_FLOOD
$IPTABLES -A FORWARD -p tcp --tcp-flags ALL NONE -j FINALIZA_FLOOD


#### Regras de repasses para o Link Internet ####
## Verifica tentativas de ataques ##
# Falsificação de endereço IP de origem no pacote (Capítulo 3, Seção 3.5) #
for REDE in $REDES_RESERVADAS; do

        $IPTABLES -A FORWARD -i $LINK -s $REDE -j FINALIZA_SPOOF

done

$IPTABLES -A FORWARD -i $LINK -s $REDE_SERVIDORES -j FINALIZA_SPOOF

## ICMP ##
# Rede de Servidores pelo Link  #
$IPTABLES -A FORWARD -i $SERVIDORES -o $LINK -s $REDE_SERVIDORES -p icmp -j
ICMP_SUSPEITA
```

```
## Tráfego Normal ##
# Servidores Internet #
$IPTABLES -A FORWARD -i $SERVIDORES -o $LINK -s $SERVIDOR_MAIL -m state --state !
INVALID -p tcp --dport 25 -j ACCEPT

$IPTABLES -A FORWARD -i $SERVIDORES -o $LINK -s $SERVIDOR_DNS_1 -m state --state !
INVALID -p udp --dport 53 -j ACCEPT

$IPTABLES -A FORWARD -i $SERVIDORES -o $LINK -s $SERVIDOR_DNS_2 -m state --state !
INVALID -p udp --dport 53 -j ACCEPT

$IPTABLES -A FORWARD -i $SERVIDORES -o $LINK -s $SERVIDOR_BETA -m state --state !
INVALID -p tcp --dport 25 -j ACCEPT


#### Regras de repasses para a Rede de Servidores ####
## Verifica tentativas de ataques ##
# Falsificação de endereço IP de origem no pacote (Capítulo 3, Seção 3.5) #
for REDE in $REDES_RESERVADAS; do

        $IPTABLES -A FORWARD -i $SERVIDORES -s $REDE -j FINALIZA_SPOOF

done

$IPTABLES -A FORWARD -i $SERVIDORES -s ! $REDE_SERVIDORES -j FINALIZA_SPOOF

## ICMP ##
# Link Internet pela Rede de Servidores #
$IPTABLES -A FORWARD -i $LINK -o $SERVIDORES -d $REDE_SERVIDORES -p icmp -j
ICMP_SUSPEITA
```

Código de configuração e das assinaturas do IDS:

```
##########################################################
#   Regras do IDS para rede Internet de Alfa      #
##########################################################


#### Definindo Variáveis ####


# Rede casa (rede que será monitorada e protegida)

var REDE_PROTEGIDA 200.254.130.0/26

# Rede externa (qualquer uma que conecte a rede protegida)
var REDE_EXTERNA any

# Servidores de e-mail
var SERVIDORES_MAIL 200.254.130.10

# Servidores de páginas
var SERVIDORES_HTTP [200.254.130.2,200.254.130.10]

# Servidores DNS
var SERVIDORES_DNS [200.254.130.2,200.254.130.10]


#### Configurando pré-processadores ####


# defrag: suporte a desfragmentação
# Capítulo 2, Seção 1.3

preprocessor frag2

# stream2: reconstrução de pacotes TCP
# Capítulo 2, Seção 3.1

preprocessor stream2: timeout 10, ports 21 80 110, maxbytes 16384

# stream4: inspeciona o estado da conexão e reconstroi pacotes TCP para
# verificação, além de detectar varreduras de portas invisíveis.
# Capítulo 2, Seção 3.1

preprocessor stream4: detect_scans
preprocessor stream4_reassemble

# http_decode: normaliza requisições HTTP convertendo caracteres do tipo %XX em
# seus tipos ASCII equivalentes visando detectar atacantes que tentam se
# esconder utilizando estes recursos misturados a requisições perigosas
# (geralmente que exploram vulnerabilidades de serviços).
# Exploração de vulnerabilidades: Capítulo 3, Seção 3

preprocessor http_decode: 80 -unicode -cginull

# bo: detector de Back Orifice
# Exploração de vulnerabilidades: Capítulo 3, Seção 3
```

```
preprocessor bo: -nobrute

# portscan: detecta variações de varredores de portas
# Protege a rede internet contra pacotes UDP e pacotes TCP (SYN) que vão em mais
# de quatro portas em menos de três segundos, inserindo um registro no arquivo
# portscan.log.
# Capítulo 3, Seção 3.4

preprocessor portscan: $REDE_PROTEGIDA 4 3 portscan.log


#### Configurando plugins de saída (registro) ####


# alert_syslog: registra alertas ao arquivo de registro do sistema “syslog”.]
# Atitude de resposta passiva. Capítulo 3, Seção 2.3.

output alert_syslog: LOG_AUTH LOG_ALERT


#### Construindo as regras para os ataques ####

## Para compactibilidade com regras internacionais e programas
# analisadores de arquivos de registros do sistema, a classificação foi mantida
# em inglês, assim como as regras. Os comentários estão em português.

## Definindo classificação dos ataques e suas prioridades ##
## Classificação é uma forma de classificar e priorizar os alertas, permitindo
## definir quais são os alertas que carregam informações mais importantes.

# A classificação é configurada no IDS da seguinte maneira:
# config classification:apelido,breve descrição,prioridade

config classification: not-suspicious,Not Suspicious Traffic,3

config classification: unknown,Unknown Traffic,3

config classification: bad-unknown,Potentially Bad Traffic, 2

config classification: attempted-recon,Attempted Information Leak,2

config classification: successful-recon-limited,Information Leak,2

config classification: successful-recon-largescale,Large Scale Information Leak,2

config classification: attempted-dos,Attempted Denial of Service,2

config classification: successful-dos,Denial of Service,2

config classification: attempted-user,Attempted User Privilege Gain,1

config classification: unsuccessful-user,Unsuccessful User Privilege Gain,1

config classification: successful-user,Successful User Privilege Gain,1

config classification: attempted-admin,Attempted Administrator Privilege Gain,1

config classification: successful-admin,Successful Administrator Privilege Gain,1
```

```
config classification: rpc-portmap-decode,Decode of an RPC Query,2

config classification: shellcode-detect,Executable code was detected,1

config classification: string-detect,A suspicious string was detected,3

config classification: suspicious-filename-detect,A suspicious filename was
detected,2

config classification: suspicious-login,An attempted login using a suspicious
username was detected,2

config classification: system-call-detect,A system call was detected,2

config classification: tcp-connection,A TCP connection was detected,4

config classification: trojan-activity,A Network Trojan was detected, 1

config classification: unusual-client-port-connection,A client was using an unusual
port,2

config classification: network-scan,Detection of a Network Scan,3

config classification: denial-of-service,Detection of a Denial of Service Attack,2

config classification: non-standard-protocol,Detection of a non-standard protocol or
event,2

config classification: protocol-command-decode,Generic Protocol Command Decode,3

config classification: web-application-activity,access to a potentially vulnerable
web application,2

config classification: web-application-attack,Web Application Attack,1

config classification: misc-activity,Misc activity,3

config classification: misc-attack,Misc Attack,2

config classification: icmp-event,Generic ICMP event,3

config classification: kickass-porn,SCORE! Get the lotion!,1


### Assinaturas de ataques ###
## As assinaturas fazem parte de um trabalho de pesquisa sobre ataques e
## publicações de assinaturas em páginas de segurança.
## A quantidade de assinaturas foi limitada pois este trabalho é apenas um
## exemplo da aplicação. As quantidades reais chegam a milhares de assinaturas.

## Assinaturas de tráfego suspeito ##


alert tcp $REDE_EXTERNA any <> $REDE_PROTEGIDA 0 (msg:"BAD TRAFFIC tcp port 0
traffic"; sid:524;  classtype:misc-activity; rev:3;)

alert udp $REDE_EXTERNA any <> $REDE_PROTEGIDA 0 (msg:"BAD TRAFFIC udp port 0
traffic"; sid:525;  classtype:misc-activity; rev:4;)
```

```
alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"BAD TRAFFIC data in TCP SYN
packet"; flags:S; dsize:>6;  sid:526;  classtype:misc-activity; rev:3;)

alert ip any any <> 127.0.0.0/8 any (msg:"BAD TRAFFIC loopback traffic";
classtype:bad-unknown; sid:528; rev:2;)

alert ip any any -> any any (msg:"BAD TRAFFIC same SRC/DST"; sameip; classtype:bad-
unknown; sid:527; rev:2;)


## Assinaturas de tentativa de exploração de vulnerabilidades em programas ##


alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA 22 (msg:"EXPLOIT ssh CRC32 overflow
/bin/sh"; flags:A+; content:"/bin/sh"; reference:bugtraq,2347; reference:cve,CVE-
2001-0144; classtype:shellcode-detect; sid:1324; rev:1;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA 22 (msg:"EXPLOIT ssh CRC32 overflow
NOOP"; flags:A+; content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90|";
reference:bugtraq,2347; reference:cve,CVE-2001-0144; classtype:shellcode-detect;
sid:1326; rev:1;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA 110 (msg:"EXPLOIT pop3 x86 linux
overflow";flags: A+; content:"|d840 cd80 e8d9 ffff ff|/bin/sh"; classtype:attempted-
admin; sid:288; rev:1;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA 143 (msg:"EXPLOIT imap
overflow";flags: A+; content:"|E8 C0FF FFFF|/bin/sh"; classtype:attempted-admin;
sid:293; rev:1;)

alert udp $REDE_EXTERNA any -> $REDE_PROTEGIDA 53 (msg:"EXPLOIT BIND Tsig Overflow
Attempt"; content:"|80 00 07 00 00 00 00 00 01 3F 00 01 02|/bin/sh";
classtype:attempted-admin; sid:314; rev:3; reference:cve,CAN-2000-0010;
reference:bugtraq,2302;)


## Assinaturas de varredores de portas ##


alert tcp $REDE_EXTERNA 10101 -> $REDE_PROTEGIDA any (msg:"SCAN myscan"; ttl: >220;
ack: 0; flags: S;reference:arachnids,439; classtype:attempted-recon; sid:613; rev:1;)

alert tcp $REDE_EXTERNA 31790 -> $REDE_PROTEGIDA 31789 (msg:"SCAN trojan hack-a-tack
probe"; content: "A"; depth: 1;  reference:arachnids,314; flags:A+;
classtype:attempted-recon; sid:614; rev:1;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"SCAN FIN"; flags: F;
reference:arachnids,27; classtype:attempted-recon; sid:621; rev:1;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"SCAN NULL";flags:0; seq:0;
ack:0; reference:arachnids,4; classtype:attempted-recon; sid:623; rev:1;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"SCAN SYN FIN";flags:SF;
reference:arachnids,198; classtype:attempted-recon; sid:624; rev:1;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"SCAN XMAS";flags:SRAFPU;
reference:arachnids,144; classtype:attempted-recon; sid:625; rev:1;)
```

```
alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"SCAN nmap fingerprint
attempt";flags:SFPU; reference:arachnids,05; classtype:attempted-recon; sid:629;
rev:1;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"SCAN NMAP XMAS";flags:FPU;
reference:arachnids,30; classtype:attempted-recon; sid:1228; rev:1;)


## Assinaturas de ataques à Servidores FTP ##


alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA 21 (msg:"FTP .forward"; content:
".forward"; flags: A+;reference:arachnids,319; classtype:suspicious-filename-detect;
sid:334; rev:2;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA 21 (msg:"FTP .rhosts";flags: A+;
content:".rhosts"; reference:arachnids,328; classtype:suspicious-filename-detect;
sid:335; rev:2;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA 21 (msg:"FTP CWD ~root"; content: "cwd
~root"; nocase; flags: A+;reference:arachnids,318; classtype:bad-unknown; sid:336;
rev:1;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA 21 (msg:"FTP satan scan";flags: A+;
content:"pass -satan"; reference:arachnids,329; classtype:suspicious-login; sid:359;
rev:2;)


## Assinaturas de ataques à Servidores de e-mail ##


alert tcp $REDE_EXTERNA 113 -> $SERVIDORES_MAIL 25 (msg:"SMTP sendmail 8.6.9
exploit";flags: A+; content:"|0a|D/"; reference:arachnids,140; reference:cve,CVE-
1999-0204; classtype:attempted-admin; sid:655; rev:1;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_MAIL 25 (msg:"SMTP exchange mime DOS";
flags: A+; content:"|63 68 61 72 73 65 74 20 3D 20 22 22|"; classtype:attempted-dos;
sid:658; rev:1;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_MAIL 25 (msg:"SMTP majordomo ifs";flags:
A+; content:"eply-to|3a| a~.`/bin/"; reference:cve,CVE-1999-0208;
reference:arachnids,143; classtype:attempted-admin; sid:661; rev:1;)


## Assinaturas de Negação de Serviço (Capítulo 3, Seção 3.6) ##


alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"DOS Land attack"; id:3868;
seq: 3868; flags:S; classtype:attempted-dos; sid:269; rev:1;)

alert udp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"DOS Teardrop attack";
id:242; fragbits:M; reference:bugtraq,124; classtype:attempted-dos; sid:270; rev:1;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA 7070 (msg:"DOS Real Audio Server";
flags: A+; content: "|fff4 fffd 06|"; reference:bugtraq,1288; reference:cve,CVE-2000-
0474; reference:arachnids,411; classtype:attempted-dos; sid:276; rev:1;)
```

```
alert udp $REDE_EXTERNA any -> $REDE_PROTEGIDA 161 (msg:"DOS Bay/Nortel Nautica
Marlin"; dsize:0; reference:bugtraq,1009; reference:cve,CVE-2000-0221;
classtype:attempted-dos; sid:279; rev:2;)

alert udp $REDE_EXTERNA any -> $REDE_PROTEGIDA 9 (msg:"DOS Ascend Route"; content:
"|4e 41 4d 45 4e 41 4d 45|"; offset: 25; depth: 50; reference:bugtraq,714;
reference:cve,CVE-1999-0060; reference:arachnids,262; classtype:attempted-dos;
sid:281; rev:2;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA 139 (msg: "DOS Winnuke attack"; flags:
U+; reference: bugtraq,2010; reference:cve,CVE-1999-0153; classtype: attempted-dos;
sid: 1257; rev:2;)


## Ataques de Negação de Serviços Distribuídos (Capítulo 3, Seção 3.6) ##


alert icmp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"DDOS TFN Probe"; id: 678;
itype: 8; content: "1234";reference:arachnids,443; classtype:attempted-recon;
sid:221; rev:1;)

alert icmp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"DDOS tfn2k icmp possible
communication"; itype: 0; icmp_id: 0; content: "AAAAAAAAAA"; reference:arachnids,425;
classtype:attempted-dos; sid:222; rev:1;)

alert udp $REDE_EXTERNA any -> $REDE_PROTEGIDA 31335 (msg:"DDOS
Trin00\:DaemontoMaster(PONGdetected)"; content:"PONG";reference:arachnids,187;
classtype:attempted-recon; sid:223; rev:1;)

alert icmp $REDE_PROTEGIDA any -> $REDE_EXTERNA any (msg:"DDOS Stacheldraht server-
response"; content: "|66 69 63 6B 65 6E|"; itype: 0; icmp_id: 667;
reference:arachnids,191; classtype:attempted-dos; sid:226; rev:1;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA 20432 (msg:"DDOS shaft client to
handler"; flags: A+; reference:arachnids,254; classtype:attempted-dos; sid:230;
rev:1;)

alert udp $REDE_EXTERNA any -> $REDE_PROTEGIDA 10498 (msg:"DDOS mstream handler to
agent"; content: "stream/"; reference:cve,CAN-2000-0138; classtype:attempted-dos;
sid:244; rev:1;)


## Assinaturas de ataques à Servidores DNS ##


alert udp $REDE_EXTERNA 53 -> $REDE_PROTEGIDA any (msg:"DNS SPOOF query response PTR
with TTL\: 1 min. and no authority"; content:"|85800001000100000000|";
content:"|c00c000c00010000003c000f|"; classtype:bad-unknown; sid:253; rev:2;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA 53 (msg:"DNS zone transfer"; content:
"|00 00 FC|"; flags: A+; offset: 13; reference:arachnids,212; classtype:attempted-
recon; sid:255; rev:2;)

alert udp $REDE_EXTERNA any -> $REDE_PROTEGIDA 53 (msg:"DNS named authors attempt";
content:"|07|authors"; offset:12; content:"|04|bind"; nocase; offset: 12;
reference:arachnids,480; classtype:attempted-recon; sid:256; rev:1;)
```

```
alert udp $REDE_EXTERNA any -> $REDE_PROTEGIDA 53 (msg:"DNS named version attempt";
content:"|07|version"; offset:12; content:"|04|bind"; nocase; offset: 12;
reference:arachnids,278; classtype:attempted-recon; sid:257; rev:1;)

alert tcp $REDE_EXTERNA any -> $REDE_PROTEGIDA 53 (msg:"DNS EXPLOIT named 8.2-
>8.2.1";flags: A+; content:"../../../../../../../../../"; reference:cve,CVE-1999-
0833; classtype:attempted-admin; sid:258; rev:1;)


## Assinaturas de ataques à Servidores HTTP (WEB) ##


alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-CGI php access";flags:
A+; uricontent:"/php.cgi"; nocase; reference:bugtraq,2250; reference:arachnids,232;
classtype:attempted-recon; sid:824; rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-CGI perl.exe
access";flags: A+; uricontent:"/perl.exe"; nocase;
reference:arachnids,219;classtype:attempted-recon; sid:832; rev:1;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-CGI uploader.exe
access";flags: A+; uricontent:"/uploader.exe"; nocase;reference:cve,CVE-1999-
0177;classtype:attempted-recon; sid:837; rev:1;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-CGI finger access";
flags: A+; uricontent:"/finger"; nocase; reference:arachnids,221; reference:cve,CVE-
1999-0612;classtype:attempted-recon; sid:839; rev:1;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-CGI edit.pl
access";flags: A+; uricontent:"/edit.pl"; nocase;classtype:attempted-recon; sid:855;
rev:1;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-CGI zsh access";flags:
A+; uricontent:"/zsh"; nocase; reference:cve,CAN-1999-0509; classtype:attempted-
recon; sid:1309; rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-CGI ksh access";flags:
A+; uricontent:"/ksh"; nocase; reference:cve,CAN-1999-0509;classtype:attempted-recon;
sid:865; rev:1;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-CGI visadmin.exe
access";flags: A+; uricontent:"/visadmin.exe"; nocase; reference:bugtraq,1808;
reference:cve,CAN-1999-1970;classtype:attempted-recon; sid:867; rev:1;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-CGI tcsh access";flags:
A+; uricontent:"/tcsh"; nocase; reference:cve,CAN-1999-0509;classtype:attempted-
recon; sid:872; rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-CGI formmail
access";flags: A+; uricontent:"/formmail"; nocase; reference:bugtraq,1187;
reference:cve,CVE-1999-0172; reference:arachnids,226; classtype:attempted-recon;
sid:884; rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-CGI bash access";flags:
A+; uricontent:"/bash"; nocase; reference:cve,CAN-1999-0509; classtype:attempted-
recon; sid:885; rev:1;)
```

```
alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-CGI phf access";flags:
A+; uricontent:"/phf"; nocase; reference:bugtraq,629; reference:arachnids,128;
reference:cve,CVE-1999-0067;  classtype:attempted-recon; sid:886; rev:3;)


## Assinaturas de ataques à Servidores Cold Fusion (Servidor WEB da Alfa) ##


alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-COLDFUSION cfcache.map
access";flags: A+; uricontent:"/cfcache.map"; nocase; reference:bugtraq,917;
reference:cve,CVE-2000-0057; classtype:attempted-recon; sid:903; rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-COLDFUSION
application.cfm access";flags: A+;
uricontent:"/cfdocs/exampleapp/publish/admin/application.cfm";
nocase;reference:bugtraq,1021; classtype:attempted-recon; sid:905; rev:1;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-COLDFUSION getfile.cfm
access";flags: A+; uricontent:"/cfdocs/exampleapp/email/getfile.cfm";
nocase;reference:bugtraq,229; classtype:attempted-recon; sid:906; rev:1;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-COLDFUSION administrator
access"; uricontent:"/cfide/administrator/index.cfm"; nocase; flags:A+;
classtype:attempted-recon; sid:908; rev:1;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-COLDFUSION admin decrypt
attempt";flags: A+; content:"CFUSION_DECRYPT()"; nocase; reference:bugtraq,550;
classtype:web-application-attack; sid:924; rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-COLDFUSION settings
refresh attempt";flags: A+; content:"CFUSION_SETTINGS_REFRESH()";
nocase;reference:bugtraq,550; classtype:web-application-attack; sid:927; rev:2;)


## Assinaturas de ataques à Servidores WEB IIS ##


alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS ..\.. access";flags:
A+; content:"|2e2e5c2e2e|"; reference:bugtraq,2218; reference:cve,CAN-1999-0229;
classtype:web-application-attack; sid:974; rev:3;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS .bat? access";flags:
A+; uricontent:".bat?&"; nocase; reference:bugtraq,2023; reference:cve,CVE-1999-0233;
classtype:web-application-activity; sid:976; rev:3;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS .cnf access";
content:".cnf"; nocase; flags:a+; classtype:web-application-activity; sid:977;
rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS ASP contents view";
flags: A+; content:"%20"; content:"&CiRestriction=none"; nocase;
content:"&CiHiliteType=Full"; nocase; reference:cve,CAN-2000-0302;
reference:bugtraq,1084; classtype:web-application-attack; sid:978; rev:4;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS File permission
canonicalization"; uricontent:"/scripts/..%c0%af../"; flags: A+; nocase;
classtype:web-application-attack; sid:981; rev:2;)
```

```
alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS MSProxy
access";flags: A+; uricontent:"/scripts/proxy/w3proxy.dll"; nocase; classtype:web-
application-activity; sid:986; rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS admin access";flags:
A+; uricontent:"/scripts/iisadmin"; nocase; classtype:web-application-attack;
sid:993; rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS cmd.exe access";
flags: A+; content:"cmd.exe"; nocase; classtype:web-application-attack; sid:1002;
rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS cmd? acess";flags:
A+; content:".cmd?&"; nocase; classtype:web-application-attack; sid:1003; rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS del attempt";flags:
A+; content:"&del+/s+c|3a|\\*.*"; nocase; classtype:web-application-attack; sid:1008;
rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS directory listing";
uricontent:"/ServerVariables_Jscript.asp"; nocase; flags:A+; classtype:web-
application-attack; sid:1009; rev:1;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS index server file
sourcecode attempt"; flags: A+; content:"?CiWebHitsFile=/";
content:"&CiRestriction=none&CiHiliteType=Full"; classtype:web-application-attack;
sid:1019; rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS perl access";flags:
A+; uricontent:"/scripts/perl"; nocase; classtype:web-application-activity; sid:1025;
rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS site server config
access";flags: A+; uricontent:"/adsamples/config/site.csc";
nocase;reference:bugtraq,256; classtype:web-application-activity; sid:1038; rev:2;)

alert tcp $SERVIDORES_HTTP 80 -> $REDE_EXTERNA any (msg:"WEB-IIS Unauthorized IP
Access Attempt"; flags: A+; content:"403"; content:"Forbidden\:"; classtype:web-
application-attack; sid:1045; rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS site/iisamples
access"; flags:A+; uricontent:"/site/iisamples"; nocase; classtype:web-application-
activity; sid:1046; rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS CodeRed v2 root.exe
access"; flags: A+; uricontent:"scripts/root.exe?"; nocase; classtype:web-
application-attack; sid: 1256; rev:2;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS outlook web dos";
flags:A+; uricontent:"/exchange/LogonFrm.asp?"; nocase; content:"mailbox="; nocase;
content:"|25 25 25|"; classtype:web-application-attack; reference:bugtraq,3223;
sid:1283; rev:4;)

alert tcp $REDE_EXTERNA any -> $SERVIDORES_HTTP 80 (msg:"WEB-IIS scripts access";
flags:A+; uricontent:"/scripts/"; nocase; classtype:web-application-activity;
sid:1287; rev:2;)
```

```
## Assinaturas de tráfego ICMP suspeito ##


alert icmp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"ICMP ISS Pinger";
content:"|495353504e475251|";itype:8;depth:32; reference:arachnids,158;
classtype:attempted-recon; sid:465; rev:1;)

alert icmp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"ICMP PING NMAP"; dsize: 0;
itype: 8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:1;)

alert icmp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"ICMP redirect
host";itype:5;icode:1; reference:arachnids,135; reference:cve,CVE-1999-0265;
classtype:bad-unknown; sid:472; rev:1;)

alert icmp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"ICMP superscan echo";
content:"|0000000000000000|"; itype: 8; dsize:8; classtype:attempted-recon; sid:474;
rev:1;)

alert icmp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"ICMP traceroute ipopts";
ipopts: rr; itype: 0; reference:arachnids,238; classtype:attempted-recon; sid:475;
rev:1;)

alert icmp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"ICMP Source Quench"; itype:
4; icode: 0; classtype:bad-unknown; sid:477; rev:1;)

alert icmp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"ICMP Broadscan Smurf
Scanner"; itype: 8; icmp_id: 0; icmp_seq: 0; dsize:4;  classtype:attempted-recon;
sid:478; rev:1;)

alert icmp $REDE_EXTERNA any -> $REDE_PROTEGIDA any (msg:"ICMP PING Sniffer
Pro/NetXRay network scan"; itype:8;
content:"|43696e636f204e6574776f726b2c20496e632e|"; depth:32; sid:484;
classtype:misc-activity; rev:2;)


## Assinaturas de comprometimento de máquinas (máquinas que respondem à ataques
## copiando arquivos remotos, listando o conteúdo de um diretório, etc.) ##

alert tcp $SERVIDORES_HTTP 80 -> $REDE_EXTERNA any (msg:"ATTACK RESPONSES http dir
listing"; content: "Volume Serial Number"; flags: A+; classtype:bad-unknown;
sid:1292; rev:1;)

alert tcp any any -> any any (msg:"ATTACK RESPONSES id check returned root";
flags:A+; content: "uid=0(root)"; classtype:bad-unknown; sid:498; rev:2;)

alert tcp $SERVIDORES_HTTP 80 -> $REDE_EXTERNA any (msg:"ATTACK RESPONSES command
completed"; content:"Command completed"; nocase; flags:A+; classtype:bad-unknown;
sid:494; rev:2;)

alert tcp $SERVIDORES_HTTP 80 -> $REDE_EXTERNA any (msg:"ATTACK RESPONSES directory
listing"; content:"Directory Listing of"; nocase; flags:A+; classtype:unknown;
sid:496; rev:2;)

alert tcp $SERVIDORES_HTTP 80 -> $REDE_EXTERNA any (msg:"ATTACK RESPONSES file copied
ok"; content:"1 file(s) copied"; nocase; flags:A+; classtype:bad-unknown; sid:497;
rev:2;)
```